Circuit Simulation and Description Library

Joseph Cordina

26th March, 2006

This is the description for the assignment unit CSA 1012, Object Oriented Programming, for the year 2007. This assignment is worth 15% of the total mark for this unit. The deadline for this project is 16th Apr, 2007 at 11:59pm. Unless you have proof of being too sick to click on submit on the web-page, late submissions will not be accepted. This is an individual assignment. Under no circumstances should code and/or design diagrams be shared with others. You will submit your code and documentation (as explained below) on the online submission system found at Joseph Cordina’s website. Unless you are already registered on the ASS system, you will need to contact the departmental administration (Mr. Vincent Sammut) for the appropriate registration ticket. Please note that you need to present your ID card to be given your ticket. Additionally you have to submit a signed copy of the plagiarism form to Mr. Vincent Sammut by Monday 16th April, 2007. Unless you do this, your assignment will not be marked. I will not accept hard copies of the assignment, so please do not leave it to the very end to submit your assignment. Please be reminded that you cannot copy and plagiarise to ease your way to a final submission. While you may discuss ideas with others, do not steal. You can find more information at http://www.cs.um.edu.mt/resources/plagiarism/

Read this information well since afterwards no excuses will save you. Anyone resorting to such methods will be considered as trying to cheat and will even risk the removal from the degree program.

1 Submission Contents

You will submit your .java or your .cs files making up the Circuit library system on the online assignment submission system. All your source code has to be compilable using the standard javac or csc command. Make sure you do not include a main() function in your code, since I will build my own to test your code. Do not encapsulate your code in packages and use standard implementations of the compilers. Also do not zip, tar or rar source files. If I cannot easily compile your code since you used some IDE, its your fault, you will not get any marks !! You will also include a brief description of how your classes interact in a file called report.txt. Additionally, you are expected to provide a file called API.txt that provides the public classes, methods and members that your library provides. If using java, you may use javadoc to generate an html file giving this API, in which case the file will be called API.html. Users of C# may use an equivalent tool. Finally you will submit a jpg file containing a UML class diagram (see below). Anyone failing to submit all of these parts might not get any marks for this assignment.

Before submitting your code, please do clean up your code. Remove any dead code and commented code. Also do place some comments to describe your structure in the code, at least to show that you understand how your own program works. Anyone wanting to use a language different from Java or C#, please let me know before hand.
2 Class Description

Whenever describing class structures, it is common to make use of UML diagrams. For this assignment, you are requested to depict your classes using the UML standard. Figure 1 depicts a simple class structure UML. It shows that class B and class C inherit from class A. It shows that a1 is an int and is a public member of A. Also that A has a constructor A() and a public function f that takes an int and returns void. Additionally it shows that A contains a collection of objects of class type D (in an array, Vector, ...). The symbol 1..* shows that A contains 1 or more of D.

You are expected to present a similar diagram to depict your class structures, filling in all the public members of each class. Note that the UML standard is able to represent much more than this diagram is showing, but this should be enough for your needs.

3 Circuit Description Library

For your assignment, you are expected to develop a library of circuit components with which the user of your library may build more complicated circuits. After building a particular circuit using your components, the programmer may simulate the circuit on a particular input and he or she may output a textual description of the circuit components.

Every circuit component will be a subclass of class CComponent. To get you started, Figure 2 shows the code for the CComponent class. Note that you have to use this class as is and you may not change its contents. The code given is in Java, you may change it to its equivalent in C#.

You are expected to develop the following circuit components at the very least. The constructor of each of these components should receive the inputs for the particular component. The constructor of the OneWire and TwoWire components should be initialised boolean objects and for the Var component, it should be a String representing the name of the variable.

- **OneWire** *(A single boolean value can be true or false)*
- **TwoWire** *(A double boolean value)*
- **Inv** *(An inverter gate (NOT))*
public abstract class CComponent
{
    protected int noOfOutputs;

    public CComponent(Integer outs)
    {
        noOfOutputs = outs.intValue();
    }

    public Integer getNoOfOutputs()
    {
        return new Integer(noOfOutputs);
    }

    public abstract CComponent simulate()
        throws UninitialisedException;

    public abstract String describe();
}

Figure 2: CComponent class; parent of all circuit components

- And (An AND gate)
- Or (An OR gate)
- Xor (An Xor gate)
- HalfAdder (A half adder, expected to be built from gates)
- FullAdder (A full adder, expected to be built from halfadders)
- Var (A generic boolean variable that can contain a boolean)

As you might have noticed, the CComponent class has a number of functions. First of all, each CComponent has to be instantiated with the number of outputs that particular component has. In terms of number of inputs, that will be defined by you for each particular component inheriting from this class. Additionally, there is a function that returns the number of outputs to the caller. Finally the two most important functions. One function is called simulate that will return a CComponent containing the boolean value after the component has been evaluated. Note that this function may throw an UninitialisedException (that you have to define). This exception will be thrown whenever one tries to simulate a Var CComponent that has not been initialised. And finally the function describe will output a description of the circuit component together with its inputs.

As an example, executing the following code
OneWire w1 = new OneWire(new Boolean(true));
OneWire w2 = new OneWire(new Boolean(false));
And a1 = new And(w1, w2);
System.out.println(a1.describe());
System.out.println(a1.simulate().describe());

should give

AND(True,False)
False

For this assignment, you can limit yourself to defining components that return up to two outputs (note that the number of inputs should be limitless). Note that as specified during lectures, one should only use class type variables for the public interface of your classes, thus your public methods should always accept or return references, and not basic data types. Those of you using C# may use the classes provided on my website, and if you wish you may also modify them for your needs.

4 Automatic Circuit Simulation

After having defined your circuit library, your next step is to perform automatic circuit simulation. You will create a class called Simulator that given a circuit will generate all the possible inputs and will then output the resultant outputs. This would greatly ease the job of the engineer who is using your library to define a particular circuit. The Var CComponent will be used to represent an input that does not need to be initialised before hand. To ease your job, the Simulator class will be passed these inputs explicitly. The Simulator class should have the following public interface:

```java
// this will reset the Simulator class to a no input situation
static public void clearInputs()
// will add a particular input variable
static public void addInput (Var v)

// will output to screen all the possible inputs values (according
// to the variables passed in addInput) and the resultant outputs
// of the passed circuit
static public void generateOutputs(CComponent circuit)
```
Thus the following code

```java
Var i1 = new Var("i1");
Var i2 = new Var("i2");
Var v1 = new Var("v1");
Var v2 = new Var("v2");

And and = new And(v1,v2);
FullAdder circ = new FullAdder(i1,i2,and);
Simulator.clearInputs();
Simulator.addInput(i1); Simulator.addInput(i2);
Simulator.addInput(v1); Simulator.addInput(v2);
Simulator.generateOutputs(circ);

System.out.println();
System.out.println(circ.describe());
```

should give the following output:\(^1\):

```
(v2,v1,i2,i1)->(Out1,Out2)

(false,false,false,false)->(false,false)
(false,false,false,true)->(true,false)
(false,false,true,false)->(true,false)
(false,false,true,true)->(false,true)
(false,true,false,false)->(false,false)
(false,true,false,true)->(true,false)
(false,true,true,false)->(true,false)
(false,true,true,true)->(false,true)
(true,false,false,false)->(false,false)
(true,false,false,true)->(true,false)
(true,false,true,false)->(true,false)
(true,false,true,true)->(false,true)
(true,true,false,false)->(true,false)
(true,true,false,true)->(false,true)
(true,true,true,false)->(false,true)
(true,true,true,true)->(true,true)
```

FULLADD(i1,i2,AND(v1,v2))

## 5 In Conclusion

The above code samples should be enough to give you an idea of what has to be done for this assignment. In fact there are a lot of hints up there of how things can be done. Please note that all code samples given should be compilable and usable ‘as is’ with the library you build. To test your library, I will build a main() function and will compile it together with your code.

\(^1\)Note that the values in the output were generated by hand and might be incorrect but you get the gist anyway.
Thus make sure that you do not include a `main()` function in the code you submit. If my code does not compile with your library you will fail your assignment.

You have to use concepts covered in the OOP lectures. Thus I am expecting your system to use OOP principles including inheritance, dynamic method dispatching and exception handling. In your code you **cannot** use the `instanceof` operator in Java or the `is` operator in C# (or its equivalent). These operators go against proper OOP design and there are always ways to make these operators redundant.

Remember that OOP is a design approach. Make sure you design your classes properly before you start coding. Understanding how things can be built is 80% of the task. The coding itself is then trivial.

As is usual for my assignments, if some things are unclear or you have some major problems, you can post your query on the discussion group at [http://groups.google.com/group/oop_2007](http://groups.google.com/group/oop_2007). You need a `gmail` account to post your queries. Just in case you do not have access to such an account, you may e-mail me at `joseph.cordina(at)um.edu.mt`. Do not e-mail me directly with queries since I will not answer. DO NOT post code or partial answers on the discussion board, since you will be penalised. The aim of the discussion board is to clear misunderstanding and not to try to find an easy way out of the assignment.

**Good luck !!**