Design Document

CSL290 Assignment 2

AC-Circuit-Solver

Kunal Parihar (2016CSJ0007) Rishabh Kumar (2016CSJ0013)

March 19, 2018

1 Overall Design

The overall design of the program consists of 5 parts -

- Lexical Analyzer - This part will do the lexical analysis of the input netlist file and generate tokens out of it.
- Parser - This part will join the tokens to create meaningful SVG code.
- Drawer Class - This class will contain functions to draw svg image.
- Circuit Physics Class - This class will contain functions to calculate voltage and current values across components of the circuit.
- Viewer Class - This class will contains function to generate the viewer and take svg file as input into it.
- SVG File - This is output of parser which acts as input to viewer class.

2 Subcomponents

The component description is as follows :-

1. **Lexical analyzer -**
   It goes through the netlist file character by character and identifies the relevant tokens and returns them to parser. We have used **flex (lex)** for lexical analysis. This class does not generate syntax errors or semantic errors, it skips the parts which does not match any syntax errors and does not generate tokens for it.

2. **Parser -**
   It takes Netlist file as input. It contains the main function so , program starts from here. It takes all the tokens returned by lexical analyzer and checks for syntax and sematics error , if errors are there then it generates error message informing about the errors. It uses the functions of Drawer class and the circuit physics class to draw the image and calculate the values.

3. **Drawer Class -**
   It contains following functions

   - **DrawResistor -** This function generates SVG code in the output file to draw a Resistor at desired positions.

- **DrawCapacitor -** This function generates SVG code in the output file to draw a Capacitor at desired positions.
- **DrawInductor -** This function generates SVG code in the output file to draw a Inductor at desired positions.
- **DrawVoltageSource -** This function generates SVG code in the output file to draw a Voltage Source at desired positions.
- **DrawCurrentSource -** This function generates SVG code in the output file to draw a Current source at desired positions.
- **DrawNet -** This function generates SVG code in the output file to draw Nets at desired positions.
- **DrawGround -** This function generates SVG code in the output file to draw a Ground (Earthing) at desired positions.

4. **Circuit Physics Class -**
   This class contains functions for complex number arithmetic like complex no. addition, multiplication.

5. **Viewer Class -**
   This class contains the functions to generate a window for viewing SVG Image. It will provide functionality to zoom and click on components.

6. **SVG File -**
   This file contains SVG code to generate SVG image.

3 Mathematics Used

Complex Arithmetic is used to calculate the required values of currents and voltages. Following are some useful mathematical equations used.

1. Add :
$$(x_1 + jy_1) + (x_2 + jy_2) = (x_1 + x_2) + j(y_1 + y_2)$$

2. Subtract :
$$(x_1 + jy_1) - (x_2 + jy_2) = (x_1 - x_2) + j(y_1 - y_2)$$

3. Multiply :
$$(x_1 + jy_1)(x_2 + jy_2) = (x_1x_2 - y_1y_2) + j(x_1y_2 + x_2y_1)$$

4. Divide :
$$(x_1 + jy_1)(x_2 + jy_2) = \frac{(x_1x_2 - y_1y_2) - j(x_1y_2 - x_2y_1)}{{x_2}^2 + {y_2}^2}$$

4 Physics Used

AC Circuit Physics is used to calculate the required values of currents and voltages. Following are some useful physics equations used.

1. Voltage across resistor :

$$V_R = I_R \sin(\omega t + 0)$$

2. Voltage across capacitor :

$$V_C = I_C \sin(\omega t - \pi/2)$$

3. Voltage across inducitor :

$$V_L = I_L \sin(\omega t + \pi/2)$$

4. Voltage in LCR Circuit :

$$V = \sqrt{V_R^2 + (V_L - V_C)^2}$$

5. Impedance of Resistor :

$$Z_R = R$$

6. Impedance of Capacitor :

$$Z_C = \frac{1}{j\omega C}$$

7. Impedance of Inductor :

$$Z_L = j\omega L$$

8. Voltage across any component :

$$V_{com} = I_{com} Z_{com}$$

9. Parallel combination of impedance :

$$Z_{net} = \frac{Z_1 Z_2}{Z_1 + Z_2}$$

10. Series combination of impedance :

$$Z_{net} = Z_1 + Z_2$$

11. **Kirchoff's Current Law** : The algebraic sum of currents in a network of conductors meeting at a point is zero.

12. **Kirchoff's Voltage Law** : The algebraic sum of the products of the resistances of the conductors and the currents in them in a closed loop is equal to the total emf available in that loop.

5 Testing

5.1 Manual Testing

- Testing through KVL and KCL throughout the circuit.
- Testing for correct circuit diagram will be done by hand.
- Testing for correctness of algorithm using print commands.
- Testing for errors like segmentation fault and stack overflow using debugger.

5.2 Automatic Testing

- Comparing the circuit and results obtained to availble online circuit solvers.
- Comparing the viewer with online available viewers.

6 Working features

1. **Zoomable Image -**
   The svg image produced by the program is zoomable and user can zoom in and out the image by clicking zoom-in(+) and zoom-out(-) buttons on the window. The image doesn't get distorted on zooming in.

2. **Scalable Circuit -**
   The circuit produced by the program is scalable in the terms of nodes. User can add as high as 1000 nodes in the circuit.

3. **Clickable Components -**
   The circuit produced by the program is clickable i.e, user can click on any component and view the voltage and current associated with it.

Upon clicking or hovering on a component, the corresponding part of the circuit gets zoomed-in and the properties of that component are displayed.

4. **Compact Circuit -**
   The created circuit will be compact i.e, if a voltage or current sour is connected to three or more components, then, program identifies this and doesn't copy the voltage or current source repeatedly.

5. **Circuit area minimization -**
   The circuit drawn by the program is visual friendly (components and nets are not messed up). The circuits seems to be close to that drawn by human.

6. **Error Generation -**
   The program generates errors like

   - *Open Circuit* - this error is generated when the specifications in the netlist file cannot create an open circuit. Although the program doesn't refrain from displaying the circuit. Voltage and current values are not calculated in this case.
   - *Short Circuit* - this error is generated when the specifications in the netlist file create a component which has no potential difference across it. Although the program doesn't refrain from displaying that component as a part of the circuit. Voltage and current values are calculated accordingly.

7. **SVG Viewer -**
   We will create our own viewer to display the svg image. The viewer will provide the zoomable and clickable features and the viewer window will be resizable also.

7 Additional features

Removable components -
The user can remove any passive component like resistor, capacitor, and inductor by selecting and deleting the component. User can then revaluate the circuit.