Developer Documentation.

Product name	: Electro-Simulator (Electronic Circuit Simulator).
Version	: 1.0
Web site	: electrosimulator.blogspot.com
Product type	: Open source
License	: Mozilla Public License (MPL)
Authors	:
	K.L. Pushpika Prasad Liyanaarachchi Dulaj Madusanka Bandara Uvidu Aroshan Peries Hasitha M. Karunathilaka
Email	: electrosimulator@gmail.com
Address	:
	Department of computer science, University of Jaffna, Jaffna, Sri Lanka.
Git URL	: https://github.com/Electro-simulator/electro.git

Project Directory Structure

In our project, when we develop this we have made some files and folders for better understanding. And for a developer, we have mentioned the structure of our project as below,

- Images

  This is a folder, this contains all the images that are used by the “index.html” file.
  Here the images that are inside of this folder.

  logo.png
  and_gate.png
  nand_gate.png
  not_gate.png
  xnor_gate.png
  xor_gate.png
  or_gate.png
  nor_gate.png

- Lib

  This folder contains JavaScript libraries.
  Here the library files that are inside of this folder.

  bootstrap.js
  boostrap.min.js
  d3.js
  d3.v3.min
  jquery-2.1.3.min.js
  live.js

- Css

  This folder contain all the css files.
  Here the css files that are inside of this folder.

  boostrap.css
  boostrap.css.map
  boostrap.min.css
  boostrap-theme.css
  boostrap-theme.css.map
  boostrap-theme.min.css
  custom.css
  index.css

- scripts

  This folder contain scripts files which used to maintain the web application.
  Here the JS files that are inside of this folder.

  config.js
  functions.js
  gateOperatorDrag.js
  gateOperators.js
  labelStreamsDrag.js
  labelStreams.js
  layout.js

Now let's have a look at the Methods and variables in JS files.

config.js

(This is defines "svg" cavers for main layout)

Variables

'operatorsRectConfig' : Define the operator display area in the web application page.
'streamsRectConfig' : Define the label stream display area in the web application page.
'drawingRectConfig' : Define the workspace area of the web application.
'resultRectConfig' : Define the result (output) tables area in the web application.
'menuRectConfig' : Define the menu area of the web application.

This each part in the SVGcanvases is defined as a "config". If some developer need to add more parts ("configs") for the area, then they have to define area "config" as follows,

```
var <CONFIG_NAME> = {
        height : <height_number>,
        width : <width_number>,
        x : <distance_from_left_side>, - for 'icons' , x => vertMargin
        y : <distance_from_top>, - for 'icons' , y => horMargin
        background : '<background_color>'
};
```

layout.js

(This JavaScript file consider about placing each configs(from config.js) into the HTML page 'index.html")

Methods

drawOperators()	: Functions are calling from function.js
drawStreams()	: Functions are calling from function.js

Variables

mainLayout	: This is use to draw the layout.
operators	: This is use to draw the gates layout.
streams	: This is use to draw the labels layout.
selectedObject	: This is use for mouse events.
selectedCircle	: This is use for select the red dotes of the operator or labels.
'mouseover'	:This means 'mouse pointer on the object'.
'mouseout'	: This means 'mouse pointer not on the object'
'mousemove'	: This means 'check dragging or not and do the actions'.
Drawings	: This is use to draw the drawing layout.
Output	: This is use to draw the output layout.
Menu	: This is use to draw the menu layout.
Gatetitle	: This is use to draw the gate_title layout.
operatorsTitle	: This is use to topic for gate_title.
labletitle	: This is use to draw the lable_title layout.
operatorsTitletxt	: This is the topic for lable_title.
workspacetitle	: This is use to draw the workspace_title layout.
workspacetitletxt	: This is the topic for workspace_title.
outputtitle	: This is use to draw the output_title layout.
outputtitletxt	: This is the topic for output_title.
menutitle	: This is use to draw the menu_title layout.
menutitletxt	: This is the topic for menu_title.

functions.js

(This is use to draw the gate operators and labels it's contain some mouse drowning functions)

Functions

addNode(nodeName, type) : This function use to create child node.
updateNode(nodeName, type, key, value) : This is use to update the graph
isDrawingLine() : This is use to check whether the line is drawing or not
getXY(mouse, current, parent) :
This function use to find the location of the mouse , current selected object and the parent object.

getCoordinatesForOperators(parentConfig, operatorConfig, noOfIcons) :
This is use to get the coordinates of the position for the gates(operators).

getConfigsForStreams(parentConfig, streamsConfig, noOfStreams) :
This is use to get the coordinates of the position for the labels(streams).
drawOperators(operators, operatorsConfig, operatorRectConfig) :
This is use to draw gate's(operator's) icons.

drawStreams(streams, streamsConfig, streamRectConfig) :
This is use to draw the label(stream)

Variables

currentLine : This is use to initialize the current selected line before it use.
circleDrag : This is use to drag the circle.

gateOperatorDrag.js

Variables

opsInWorkArea : This is use to count the gates in the workspace.
moveGroup : This is use to move gates set.
moveEnd : This is use to define gates list.
operatorBox : Gives the outline width and height and image of the current gates(operators).
operatorDrag : This is use to gate drag operations.

gateOperators.js

(This is use to define gate list)

Variables

and_gate : This is use to define and gate as a variable.
or_gate : This is use to define or gate as a variable.
nand_gate : This is use to define nand gate as a variable.
nor_gate : This is use to define nor gate as a variable.
xnor_gate : This is use to define xnor gate as a variable.
xor_gate : This is use to define xor gate as a variable.
not_gate : This is use to define not gate as a variable.
operator : This is use to get array of the operator list.

To add new operator we need to define operator as follows,

```
var <OPERATOR_NAME> = {
                        type : '<TYPE OF THE OPERATOR>',
                        icon : '<IMAGE NAME.IMAGE TYPE>'
};
```

labelStreamsDrag.js

(This is for dragging operations of the labels and for linking the labels.)

Variables

moveStream :
Take the selected object(each label) and transform the object in x+50(pixel) and y+50(pixel).

moveStreamEnd :
Label drag and drop, then do drag in the workspace.

drawingGroup :
Menu for the label stream.

r - lable :
This is for auto drawing the label after the dragging.

labelStreams.js

(This is use to define input output labels)

Variables

To add new label, we need to define labels as follows

```
var <LABEL_NAME> = {
        connectionName :  '<LABEL_NAME>', - No white space need to add between two words.
                          for space developer can use "_".
        connectionType   :  '<TYAPE_OF_THE_LABEL>' - Mostly this may be "Steam".
};
```

InputLabel_1	: This is use to define label 1 as variable.
InputLabel_2	: This is use to define label 2 as variable.
InputLabel_3	: This is use to define label 3 as variable.
InputLabel_4	: This is use to define label 4 as variable.
OutputLabel	: This is use to define Output Labels variable.
Streams	: This is use to get array of the labels list.

This is the main HTML file in our project. Here we have describe some basic functions to get a good understanding about our codes.

Index.html

Functions

clickFunction(e)	: Output Result to display output.
createTableStream	: Create tables in menu layout.
createFormStream(targetId, streamName)	: Create menu in menu layout.