

# Final Report on Computer Graphics Project: Logic Gates Simulator

Nootan Ghimire (BCT-069-21), Nishan Bajracharya (BCT-069-20)

February 27, 2015

## **Preface**

For partial fulfillment of the Computer Graphics Course, We, students of Computer Engineering Faculty, have submitted this document as a report of the details of the project. The title of the project was “logicsim-js”. We used technologies such as “javascript”, “node-js”, “jQuery”, “html” and “css” to complete the project. We also used git version controlling as source control system.

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Background . . . . .	1
1.2	Objectives . . . . .	1
<b>2</b>	<b>Methodology</b>	<b>1</b>
2.1	Technologies Used . . . . .	1
2.2	Backend Algorithms . . . . .	2
2.2.1	Basic Operations . . . . .	2
2.2.2	Loader . . . . .	3
2.2.3	Helpers . . . . .	4
2.2.4	Exceptions . . . . .	6
2.2.5	Process . . . . .	6
2.2.6	Browserify Entry . . . . .	9
2.2.7	Building the Backend and Example . . . . .	9
2.2.8	Example . . . . .	9
2.3	Front End Algorithms . . . . .	10
2.3.1	Creating Shapes . . . . .	10
2.3.2	The Shapes Module . . . . .	13
2.3.3	Main . . . . .	14
2.3.4	UI Styling . . . . .	22
2.3.5	The Main HTML File . . . . .	23
<b>3</b>	<b>Project Timeline</b>	<b>25</b>
	<b>References</b>	<b>26</b>

# 1 Introduction

## 1.1 Background

Digital Circuit Design, till date is done using hardware descriptor language. Modules are coded in any HDL and then synthesized on different inputs on different times. We found one “digital” circuit synthesizer named **Active-HDL** [1], but that was too hard for a beginner to start off with. There were other easier online browser-based tools such as Schemit.[2]. But it’s only used to graphically create circuit diagrams. These lacked the necessary simulation. Moreover, Other graphical softwares does simulations but they are based on general electronic circuit design rather than logical component design.

Thus, in order to fulfill the lack of the aforementioned type of software we designed “logicsim-js” - An Logic Gates Simulator

## 1.2 Objectives

The objectives of this project is to:

1. To simplify the process of designing Digital Modules
2. To be able to graphically *code* HDL

# 2 Methodology

We mainly used Javascript, and HTML5 to accomplish this project.

## 2.1 Technologies Used

We can list the various technology we utilized while doing this project.

1. nodejs : To code the backend
2. fabric-js : To implement the drag-drop features
3. browserify-js : To convert the backend written in CommonJS (nodejs) to Browser Javascript
4. browserify-fs : A Javascript Module that adds local filesystem support in browsers using IndexedDBs.
5. git : Source Control
6. Github : Code Hosting

## 2.2 Backend Algorithms

In this part we discuss on various algorithms that were used in order to code the software. We mostly utilized the asynchronous nature of javascript while coding different parts of the software. We used nodejs so that the backend would be more modular as nodejs allows us to separate the logics in different files and provides an easy way to glue them together.

### 2.2.1 Basic Operations

The following code lists the basic operations (basic gates) that operates on any number of inputs.

```
1 var exceptions = require('./exceptions.js')
2
3 var basic_and = function (input) {
4   //Applies 'and' to all the values in the array and returns
   value.
5   return input.reduce (function and_fn (prevValue, curValue,
   idx, arr) {
6     return prevValue && curValue;
7   });
8 }
9
10 var basic_or = function (input) {
11   //Applies 'or' to all the values in the array and return
   value.
12   return input.reduce (function or_fn (prevValue, curValue,
   idx, arr) {
13     return prevValue || curValue ;
14   });
15 }
16
17 var basic_not = function (input) {
18   if (input.length !== 1) {
19     throw new exception("Incorrect Number of Input!");
20   }
21   //TODO: Is input really bool?
22   return !input;
23 }
24
25 var basic_xor_two = function (a, b) {
26   //Returns xor of two number. (JS does not have ^)
27   return ( a || b ) && !( a && b );
28 }
29
30 var basic_xor = function(input){
31   //Applies 'xor' to all the values in array.
```

```

32     return input.reduce(function xor_fn(prevValue, curValue,
33         idx, arr){
34         return basic_xor_two(prevValue, curValue);
35     })
36 }
37 module.exports.basic_and = basic_and;
38 module.exports.basic_or = basic_or;
39 module.exports.basic_not = basic_not;
40 module.exports.basic_xor = basic_xor;

```

## 2.2.2 Loader

The following code allows us to load the required file from filesystem (server filesystem in CommonJS and emulated-filesystem in case of browser).

```

1 //loader.js -- loads external module
2
3
4 //var fs = require('browserify-fs');
5 var fs = require('fs');
6 var helpers = require('./helpers.js');
7
8 var defaultPath = "./example_modules/";
9
10 var load = function(moduleName, callback,
11     optionalModuleRootPath) {
12     // var isComplete = false;
13     // var jsonData = false;
14     optionalModuleRootPath = optionalModuleRootPath ||
15     defaultPath;
16     fs.readFile(optionalModuleRootPath + moduleName + '.mod',
17     'utf-8',
18     function (err, data) {
19         //console.log(data);
20         var json;
21         if ((json = helpers.tryParseJSON(data)) === false) {
22             //Fallback
23             //callback("Cannot Parse", false)
24             callback(true, "Cannot Load");
25         } else {
26             callback(false, json); //Callback with json_data
27             //jsonData = json;
28             //isComplete = true;
29         }
30     });
31
32     // var data = fs.readFileSync(optionalModuleRootPath +
33     moduleName +

```

```

31     // '.mod').toString();
32 };
33
34 /*
35 load("adder", function (data) {
36     console.log(data);
37 });*/
38
39 module.exports.load = load;
40 module.exports.fs = fs; //FOR DEBUG

```

### 2.2.3 Helpers

All the helper (backend) functions are coded in this code.

```

1 var util = require('util');
2 var Exceptions = require('./exceptions.js')
3
4 var mapMultipleArrayToObj = function (arrayOfArrays,
5     defaultValue) {
6     //console.log (util.inspect (arrayOfArrays, false, null) );
7     var returnObj = {};
8     for (key_outer in arrayOfArrays) {
9         var array = arrayOfArrays[key_outer];
10        for (key_inner in array) {
11            returnObj[array[key_inner]] = defaultValue || false;
12        }
13    }
14    return returnObj;
15 }
16
17 var mapArrayToArrayUsingObj = function (arrayOfKeys,
18     objectToUse) {
19     var returnArr = [];
20     for (var i = arrayOfKeys.length - 1; i >= 0; i--) {
21         if (objectToUse[arrayOfKeys[i]] === undefined) {
22             throw new Exceptions.VariableNotFoundException("
23                 Unknown Variable: " +
24                 arrayOfKeys[i]);
25         }
26         returnArr.push(objectToUse[arrayOfKeys[i]]);
27     };
28     return returnArr;
29 }
30
31 var changeZerosToFalse = function(arr){
32     return arr.map(function(prevVal, curVal, idx, ar){
33         return prevVal == 0 ?

```

```

32     false:
33     true;
34 });
35 }
36
37 var findModulesInArray = function(arr, name){
38     for (var i = arr.length - 1; i >= 0; i--) {
39         if (arr[i].name === name) {
40             return arr[i];
41         }
42     };
43     return false;
44 }
45
46 /**
47  * Tries to Parse JSON
48  * @param {String} jsonString String to Parse
49  * @return {boolean}         false if cannot parse
50  * @return {object}         the parsed JSON Object
51  *
52  * @copied from: http://stackoverflow.com/a/20392392/1306046
53  */
54 var tryParseJSON = function (jsonString) {
55     try {
56         var o = JSON.parse(jsonString);
57
58         // Handle non-exception-throwing cases:
59         // Neither JSON.parse(false) or JSON.parse(1234)
60         // throw errors, hence the type-checking,
61         // but... JSON.parse(null) returns 'null', and typeof
62         // null === "object",
63         // so we must check for that, too.
64         if (o && typeof o === "object" && o !== null) {
65             return o;
66         }
67     }
68     catch (e) { }
69
70     return false;
71 };
72
73 module.exports.mapMultipleArrayToObj = mapMultipleArrayToObj;
74 module.exports.mapArrayToArrayUsingObj =
75     mapArrayToArrayUsingObj;
76 module.exports.changeZerosToFalse = changeZerosToFalse;
77 module.exports.tryParseJSON = tryParseJSON;

```



## 2.2.4 Exceptions

The following code snippet contains a basic module for managing exceptions.

```
1 var Exceptions = {
2   InputLengthErrorException: function (message){
3     this.InputLengthErrorException.message = message;
4     this.InputLengthErrorException.type = "
      InputLengthErrorException";
5   },
6   OutputLengthErrorException: function (message) {
7     this.OutputLengthErrorException.message = message;
8     this.OutputLengthErrorException.type = "
      OutputLengthErrorException";
9   },
10  VariableNotFoundException: function (message) {
11    this.VariableNotFoundException.message = message;
12    this.VariableNotFoundException.type = "
      VariableNotFoundException";
13  },
14  ModulesNotAvailableException: function (message) {
15    this.ModulesNotAvailableException.message = message;
16    this.ModulesNotAvailableException.type = "
      ModulesNotAvailableException";
17  }
18 }
19 };
20
21 module.exports = Exceptions;
```

## 2.2.5 Process

The overall backend was then covered under a single module called “process”. The code is given below.

```
1 var basicOperations = require('./basic_op.js');
2 var Exceptions = require('./exceptions.js');
3 var helpers = require('./helpers.js');
4 var loader = require('./loader.js');
5 //Debug
6 var util = require('util');
7
8
9 module.exports.doProcess = function proc (object, inputs,
10   callback){
11   //var currentObjInstance = new objInternalInstance(object.
12     name);
13   var returnOutputs = [];
14   //currentObjInstance.returnWires = {};
```

```

13  var variables = helpers.mapMultipleArrayToObj([
14      object.inputs,
15      object.outputs,
16      object.wires
17  ]);
18
19  //parentInstance = parentInstance || false;
20
21  //console.log("Inputs given in " + object.name + " : " +
22      inputs);
23  //console.log("Real Inputs: " + object.inputs)
24  if(inputs.length !== object.inputs.length) {
25      callback(true, {msg:"Input Length Mismatch"});
26  }
27
28  //Assign Inputs to currentObjInstance
29  for (var i = inputs.length - 1; i >= 0; i--) {
30      /*currentObjInstance*/variables[object.inputs[i]] =
31          inputs[i];
32  };
33
34  var that = this;
35  object.operations.forEach(function (currentValue, index,
36      arr) {
37      //console.log(variables);
38      if (currentValue.type === 'basic') {
39          if(currentValue.outputs.length > 1){
40              var e_msg = parentInstance === false ?
41                  "Error in Operation: " + currentValue.instance +
42                  "More than one output specified." :
43                  "Error in Operation: " + currentValue.instance +
44                  "Called From: " + parentInstance.identifier;
45              callback(true, {msg:e_msg});
46          } else {
47              /*currentObjInstance*/variables[currentValue.outputs
48                  [0]] =
49                  basicOperations[currentValue.instance](
50                      helpers.mapArrayToArrayUsingObj(currentValue.inputs
51                          ,
52                          /*currentObjInstance*/variables));
53              if(index === arr.length - 1){
54                  //console.log(variables);
55                  //console.log(variables)
56                  callback(false, returnOutputArrayVal(object.
57                      outputs, variables));
58              }
59          }
60      } else { /* if not basic */

```

```

55     loader.load(currentValue.instance, function (err, data)
56         {
57         if(err){
58             callback(true, {msg:"Cannot Load Module" +
59                 currentValue.instance});
60         } else {
61             //Yay! Data
62             //console.log(data);
63             proc(data, helpers.mapArrayToArrayUsingObj(
64                 currentValue.inputs,
65                 /*currentObjInstance.*/variables),
66                 function (error, output) {
67                     if(error === true){
68                         callback(true, {msg:"Error processing: "+
69                             currentValue.instance,
70                                 error_details:output});
71                     } else {
72                         //Yay!
73                         //Output returns array?single
74                         //if(currentObjInstance.returnOutputs)
75                         //console.log("This:" +output);
76                         for (var i = currentValue.outputs.length - 1;
77                             i >= 0; i--) {
78                             variables[currentValue.outputs[i]] = output
79                                 [i]
80                         };
81                         //console.log(currentValue);
82                         if(index === arr.length - 1){
83                             //console.log(variables);
84                             //console.log(variables)
85                             callback(false, returnOutputArrayVal(object
86                                 .outputs, variables));
87                         }
88                     }
89                 });
90             }
91         });
92     }
93 }
94
95 function returnOutputArrayVal(objOutputArray, variables){
96     var r0 = [];
97     for (var j = objOutputArray.length - 1; j >= 0; j--) {
98         r0[j] = variables[objOutputArray[j]];
99     };
100     return r0;
101 }

```

## 2.2.6 Browserify Entry

Now all the required APIs were exposed to the browser with the help of browserify-js and a file browserify\_entry.js. The contents are shown below.

```
1 //var fs = require('browserify-fs');
2 module.exports.loader = require('./loader');
3 module.exports.process = require('./process.js');
```

## 2.2.7 Building the Backend and Example

Open loader.js and change `require('fs')` to `require('browserify-fs')`.

```
1 $ browserify browserify_entry.js --standalone LogicSim -o ../
  build/logicsim-0.2.js
```

Then include **logicsim-0.2.js** in your html.

## 2.2.8 Example

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <title>LogicSim JS 0.2</title>
6 </head>
7 <body>[]
8   Check Console
9   <button onclick="appendData();">Add Sample Data to
    current FileSystem</button>
10  <button onclick="doProcess();">Do the process!</button>
11 </body>
12 <script src="../../build/logicsim-0.2.js" type="text/
    javascript"></script>
13 <script type="text/javascript">
14   var loader = LogicSim.loader;
15   var process = LogicSim.process;
16   function doProcess(){
17     var data = loader.load("adder", function (err, data) {
18       if (err){
19         console.log("Oops! Error!");
20         console.log(data);
21       } else {
22         process.doProcess(data, [true, false, true], function
          (error, output){
23           if (error === true) {
24             console.log("Oops! Error!");
25             console.log(output);
26           } else {
```

```

27         console.log(output);
28     }
29     });
30 }
31 });
32 }
33
34 function appendData(){
35     var fs = loader.fs;
36     fs.mkdir('./example_modules', function(e,d){
37         console.log("DIR Error: Means it's already there if not
38             undefined " + e); console.log("Dir Data?" + d);
39         fs.writeFile('./example_modules/adder.mod', '{ "name":
40             "fulladder", "inputs": [ "A", "B", "Ci" ], "outputs"
41             : [ "S", "Co" ], "wires": [ "w1", "w2", "w3" ], "
42             operations": [ { "instance": "basic_xor", "type": "
43             basic", "inputs": [ "A", "B", "Ci" ], "outputs": [ "
44             S" ] }, { "instance": "basic_and", "type": "basic",
45             "inputs": [ "A", "B" ], "outputs": [ "w1" ] }, { "
46             instance": "basic_and", "type": "basic", "inputs": [
47             "Ci", "A" ], "outputs": [ "w2" ] }, { "instance": "
48             basic_and", "type": "basic", "inputs": [ "Ci", "B"
49             ], "outputs": [ "w3" ] }, { "instance": "basic_or",
50             "type": "basic", "inputs": [ "w1", "w2", "w3" ], "
51             outputs": [ "Co" ] } ] }',
52         function(e,d){
53             console.log("FILE Error: Could mean it's already
54                 there if not undefined" + e);
55             console.log("file Data? Maybe? "+d);
56         });
57     });
58 }
59 </script>
60 </html>

```

## 2.3 Front End Algorithms

### 2.3.1 Creating Shapes

```

1 aCreate = document.getElementById("aCreate");
2 oCreate = document.getElementById("oCreate");
3 nCreate = document.getElementById("nCreate");
4 iCreate = document.getElementById("iCreate");
5 lCreate = document.getElementById("lCreate");
6 cClear = document.getElementById("cClear");
7
8 aCreate.onclick = function() {
9     v = shapeAnd("and_" + aNum, 100 + Math.random() * (canvas
10         .width - 200), 100 + Math.random() * (canvas.height -

```

```

        200))
10     mainShapes["and_" + aNum] = {
11         shape: v
12     }
13     aNum++;
14 }
15
16 oCreate.onclick = function() {
17     v = shapeOr("or_" + oNum, 100 + Math.random() * (canvas.
        width - 200), 100 + Math.random() * (canvas.height -
        200))
18     mainShapes["or_" + oNum] = {
19         shape: v
20     }
21     oNum++;
22 }
23
24 nCreate.onclick = function() {
25     v = shapeNot("not_" + nNum, 100 + Math.random() * (canvas
        .width - 200), 100 + Math.random() * (canvas.height -
        200))
26     mainShapes["not_" + nNum] = {
27         shape: v
28     }
29     nNum++;
30 }
31
32 iCreate.onclick = function() {
33     var inp = new fabric.IText('0', {
34         fontFamily: 'Source Sans Pro',
35         left: 100 + Math.random() * (canvas.width - 100),
36         top: 100 + Math.random() * (canvas.height - 100),
37         fontSize: 20,
38         hasControls: false,
39         transparentCorners: false,
40         padding: 5,
41         shapeType: "input",
42         type: "i-number",
43         dataType: "input",
44         linesIn: 0,
45         hasBorders: true,
46         outputObj: {},
47         output: 0,
48         input: {}
49     });
50     inp.set({
51         id: "ip_" + iNum
52     });
53     inp.on('editing:entered', function() {

```

```

54     onkeydown = function(e) {
55         e.preventDefault();
56         if (e.keyCode == 48) {
57             inp.setText("0");
58             inp.output = 0;
59         }
60         if (e.keyCode == 49) {
61             inp.setText("1");
62             inp.output = 1;
63         }
64         canvas.renderAll();
65     }
66 }
67 if (waitingForLine) inp.selectable = false;
68 mainInputs["ip_" + iNum] = {
69     shape: inp
70 };
71 iNum++;
72 canvas.add(inp);
73 }
74
75 lCreate.onclick = function() {
76     if (!waitingForLine) {
77         waitingForLine = true;
78         canvas.forEachObject(function(o) {
79             o.selectable = false;
80         });
81         lCreate.setAttribute("class", "menu-selected");
82     } else {
83         waitingForLine = false;
84         canvas.forEachObject(function(o) {
85             if (o.dataType != "line") o.selectable = true;
86             o.set({
87                 "stroke": "#666",
88                 "strokeWidth": 1
89             });
90         });
91         closeArea.set({
92             "stroke": "#fff"
93         });
94         canvas.renderAll();
95         canvas.selectable = false;
96         lCreate.removeAttribute("class");
97     }
98 }
99
100 cClear.onclick = function() {
101     removeAllObjects();
102 }

```

### 2.3.2 The Shapes Module

```
1 function shapeAnd(id, x, y, size) {
2   if (size === undefined) size = 50;
3   var path = "M0,0L50,0A50,50 0 0,1 50,100L0,100Z";
4   var andShape = new fabric.Path(path);
5   andShape.set({
6     fill: "#fff",
7     stroke: "#666",
8     id: id,
9     width: 100,
10    height: 100,
11    top: y,
12    left: x,
13    shapeType: "and",
14    hasControls: false,
15    scaleX: size / 50,
16    scaleY: size / 50,
17    dataType: "shape",
18    linesIn: 0,
19    hasBorders: false,
20    outputObj: {},
21    output: 0,
22    input: {}
23  })
24  if (waitingForLine) andShape.selectable = false;
25  canvas.add(andShape);
26  canvas.renderAll();
27  canvas.setActiveObject(andShape);
28  return andShape;
29 }
30
31 function shapeOr(id, x, y, size) {
32   if (size === undefined) size = 50;
33   var path = "M0,0Q150,50,0,100M0,0Q50,50,0,100";
34   var orShape = new fabric.Path(path);
35   orShape.set({
36     fill: "#fff",
37     stroke: "#666",
38     id: id,
39     width: 100,
40     height: 100,
41     top: y,
42     left: x,
43     shapeType: "or",
44     hasControls: false,
45     scaleX: size / 50,
46     scaleY: size / 50,
47     dataType: "shape",
```



```

48     linesIn: 0,
49     hasBorders: false,
50     outputObj: {},
51     output: 0,
52     input: {}
53   })
54   if (waitingForLine) orShape.selectable = false;
55   canvas.add(orShape);
56   canvas.renderAll();
57   canvas.setActiveObject(orShape);
58   return orShape;
59 }
60
61 function shapeNot(id, x, y, size) {
62   if (size === undefined) size = 50;
63   var path = "M0,0L75,50L0,100ZM75,50C75,43,85,43,85,50M75
64             ,50C75,57,85,57,85,50M75,50C75,43,85,43,85,50M75,50C75
65             ,57,85,57,85,50";
66   var notShape = new fabric.Path(path);
67   notShape.set({
68     fill: "#fff",
69     stroke: "#666",
70     id: id,
71     width: 100,
72     height: 100,
73     top: y,
74     left: x,
75     shapeType: "not",
76     hasControls: false,
77     scaleX: size / 50,
78     scaleY: size / 50,
79     dataType: "shape",
80     linesIn: 0,
81     hasBorders: false,
82     outputObj: {},
83     output: 0,
84     input: {}
85   })
86   if (waitingForLine) notShape.selectable = false;
87   canvas.add(notShape);
88   canvas.renderAll();
89   canvas.setActiveObject(notShape);
90   return notShape;
91 }

```

### 2.3.3 Main

```

1 //Initial Values
2 aNum = 0;

```

```

3  oNum = 0;
4  nNum = 0;
5  iNum = 0;
6  lNum = 0;
7  waitingForLine = false;
8  lines = {};
9  mainShapes = {};
10 mainInputs = {};
11 canvas = new fabric.Canvas("canvas");
12
13 window.onload = function() {
14     var w = window,
15         d = document,
16         e = d.documentElement,
17         g = d.getElementsByTagName('body')[0],
18         wid = w.innerWidth || e.clientWidth || g.clientWidth,
19         hei = w.innerHeight || e.clientHeight || g.
                clientHeight;
20
21     var header = document.getElementsByTagName('header')[0];
22     var head_h = header.offsetHeight;
23
24     //Canvas Properties
25     canvas.backgroundColor("#fff");
26     canvas.setWidth(wid);
27     canvas.setHeight(hei - head_h);
28     canvas.selectable = false;
29
30     //Area on the right of the canvas where objects are
        dragged to be deleted
31     closeArea = new fabric.Rect({
32         width: 100,
33         height: canvas.height,
34         left: canvas.width - 100,
35         top: 0,
36         selectable: false,
37         id: "closeArea",
38         opacity: 0,
39         fill: "#f22"
40     })
41     canvas.add(closeArea);
42     canvas.renderAll();
43 }
44
45 canvas.on('mouse:down', function(o) {
46     if (waitingForLine) { //In line drawing mode
47         canvas.forEachObject(function(o) {
48             o.set({
49                 "stroke": "#666",

```

```

50         "strokeWidth": 1
51     })
52 })
53 canvas.renderAll();
54 canvas.selectable = false;
55 if (o.target !== undefined) {
56     if (o.target.hasOwnProperty("shapeType")) {
57         //Select shape based on type : logic shape or
58         //logic input
59         if (o.target.dataType == "shape") {
60             getShape = mainShapes;
61         } else {
62             getShape = mainInputs;
63         }
64         //Select source of line *
65         curr = o.target.id;
66         getShape[o.target.id].shape.set({
67             "stroke": "#49c",
68             "strokeWidth": 2
69         });
70         canvas.renderAll();
71         if (!lines.hasOwnProperty(o.target.id)) {
72             lines[o.target.id] = {
73                 to: {}
74             }
75         }
76     }
77 } else { //Not in line drawing mode
78     if (o.target !== undefined) {
79         if (o.target.hasOwnProperty("id") && o.target.id
80             != "closeArea") {
81             //On object select, set border of object to
82             //thickness 2 and color blue to denote
83             //selection
84             o.target.set({
85                 "stroke": "#49c",
86                 "strokeWidth": 2
87             });
88             canvas.renderAll();
89         }
90     }
91     closeArea.set({
92         opacity: 0,
93         selectable: false
94     });
95     canvas.renderAll();
96 }

```

```

95
96 canvas.on('mouse:up', function(o) {
97     if (waitingForLine) {
98         if (o.target !== undefined) {
99             if (o.target.hasOwnProperty("id") && o.target.id
100                != "closeArea") {
101                 //Select destination of line *
102                 o.target.set({
103                     "stroke": "#49c",
104                     "strokeWidth": 2
105                 });
106                 o.target.linesIn++;
107                 canvas.renderAll();
108                 lines[curr].to[o.target.id] = true;
109                 renderLines();
110             }
111         }
112         for (i in mainInputs) {
113             mainInputs[i].shape.selectable = false;
114             canvas.renderAll();
115         }
116     } else {
117         if (o.target !== undefined) {
118             if (o.target.id != "closeArea") {
119                 //If object is line, delete line
120                 if (o.target.isLine) {
121                     delete lines[o.target.from].to[o.target.
122                        to]
123                     canvas.remove(o.target);
124                 }
125                 //If object is at the right end of the canvas
126                 , delete the object
127                 if (o.target.left > (canvas.width - 200)) {
128                     if (lines.hasOwnProperty(o.target.id)) {
129                         for (i in lines[o.target.id].to) {
130                             canvas.remove(lines[o.target.id].
131                                to[i]);
132                         }
133                         delete lines[o.target.id];
134                     }
135                     for (i in lines) {
136                         canvas.remove(lines[i].to[o.target.id
137                            ]);
138                         delete lines[i].to[o.target.id];
139                     }
140                     canvas.remove(o.target);
141                     delete mainShapes[o.target.id];
142                     delete mainInputs[o.target.id];

```

```

139         renderLines();
140     }
141     o.target.set({
142         "stroke": "#666",
143         "strokeWidth": 1
144     })
145     canvas.renderAll();
146 }
147 }
148 }
149 closeArea.set({
150     opacity: 0,
151     selectable: false
152 });
153 canvas.renderAll();
154 })
155
156 canvas.on('object:moving', function(o) {
157     if (o.target.left > (canvas.width - 200)) { //Highlight
158         right end of canvas if object is moved there
159         o.target.set({
160             "stroke": "#e22",
161             "strokeWidth": 2
162         })
163         closeArea.set({
164             opacity: 0.6,
165             selectable: false
166         });
167         canvas.renderAll();
168     } else { //Dragged object is highlighted
169         o.target.set({
170             "stroke": "#49c",
171             "strokeWidth": 2
172         })
173         closeArea.set({
174             opacity: 0,
175             selectable: false
176         });
177         canvas.renderAll();
178     }
179     canvas.renderAll();
180     renderLines();
181 })
182
183 function renderLines() {
184     canvas.renderAll();
185     for (i in lines) {

```

```

186     if (mainShapes.hasOwnProperty(i)) source = mainShapes
187         [i];
188     else source = mainInputs[i];
189     for (j in lines[i].to) {
190         if (mainShapes.hasOwnProperty(j)) destination =
191             mainShapes[j];
192         else destination = mainInputs[j];
193         drawLine(source, destination);
194     }
195 }
196
197 function drawLine(s, d) {
198     if (s.hasOwnProperty("shape") && d.hasOwnProperty("shape"
199         )) {
200         if (s.shape.id != d.shape.id) {
201             d.shape.input[s.shape.id] = s.shape.output;
202             //Shape general positioning
203             //Source
204             sl = s.shape.left + s.shape.width;
205             st = s.shape.top + s.shape.height / 2;
206             //Destination
207             dl = d.shape.left;
208             if (d.shape.shapeType == "not" || d.shape.
209                 shapeType == "input") {
210                 dt = d.shape.top + d.shape.height * 0.5;
211             } else {
212                 if (s.shape.top + s.shape.height / 2 < d.
213                     shape.top + d.shape.height / 3) {
214                     dtPos = -0.3;
215                 } else if (s.shape.top + s.shape.height / 2 >
216                     d.shape.top + d.shape.height * 2 / 3) {
217                     dtPos = +0.3;
218                 } else {
219                     dtPos = 0;
220                 }
221                 dt = d.shape.top + d.shape.height * (0.5 +
222                     dtPos);
223             }
224             //Shape specific positioning
225             slComp = 0;
226             stComp = 0;
227             dlComp = 0;
228             dtComp = 0;
229             if (s.shape.shapeType == "not") {
230                 slComp = -(s.shape.width * 0.15);
231             }
232             if (s.shape.shapeType == "or") {

```

```

228         slComp = -(s.shape.width * 0.25);
229     }
230     if (d.shape.shapeType == "or") {
231         if (dtPos == 0) {
232             dlComp = (d.shape.width * 0.25);
233         } else {
234             dlComp = (d.shape.width * 0.16)
235         }
236     }
237     if (s.shape.shapeType == "input") {
238         slComp = 7;
239         stComp = 1;
240     }
241     if (d.shape.shapeType == "input") {
242         dlComp = -6;
243         dtComp = 1;
244     }
245     //Positioning the start point of line
246     if ((sl + slComp) > (dl + dlComp)) x1 = dl +
        dlComp;
247     else x1 = sl + slComp;
248     if ((st + stComp) > (dt + dtComp)) xt = dt +
        dtComp;
249     else xt = st + stComp;
250     l = new fabric.Line([sl + slComp, st + stComp, dl
        + dlComp, dt + dtComp], {
251         stroke: "#666",
252         left: x1,
253         top: xt,
254         selectable: false,
255         id: "ln_" + lNum,
256         dataType: "line",
257         from: s.shape.id,
258         to: d.shape.id,
259         isLine: true
260     })
261     canvas.remove(lines[s.shape.id].to[d.shape.id]);
262     canvas.add(l);
263     lines[s.shape.id].to[d.shape.id] = l;
264 }
265 }
266 }
267
268 function removeAllObjects() {
269     lines = {};
270     mainShapes = {};
271     mainInputs = {};
272     canvas.forEachObject(function(o) {
273         canvas.remove(o);

```

```

274     })
275 }
276
277 function compile() {
278     op = 0;
279     for (i in mainShapes) {
280         //console.log(mainShapes[i].shape.input);
281         for (d in mainShapes[i].shape.input) {
282             if (mainShapes.hasOwnProperty(d)) {
283                 gShape = mainShapes;
284             } else {
285                 gShape = mainInputs;
286             }
287             mainShapes[i].shape.input[d] = gShape[d].shape.
                output;
288         }
289         if (mainShapes[i].shape.shapeType == "and") {
290             mainShapes[i].shape.output = 1;
291             for (var da in mainShapes[i].shape.input) {
292                 if (mainShapes[i].shape.input[da] == 0) {
293                     mainShapes[i].shape.output = 0;
294                     break;
295                 }
296             }
297         }
298         if (mainShapes[i].shape.shapeType == "or") {
299             mainShapes[i].shape.output = 0;
300             for (var da in mainShapes[i].shape.input) {
301                 if (mainShapes[i].shape.input[da] == 1) {
302                     mainShapes[i].shape.output = 1;
303                     break;
304                 }
305             }
306         }
307         if (mainShapes[i].shape.shapeType == "not") {
308             for (var da in mainShapes[i].shape.input) {
309                 if (mainShapes[i].shape.input[da] == 0) {
310                     mainShapes[i].shape.output = 1;
311                 } else {
312                     mainShapes[i].shape.output = 0;
313                 }
314             }
315         }
316         op = (mainShapes[i].shape.output);
317         if (lines.hasOwnProperty(i)) {
318             for (z in lines[i].to) {
319                 if (mainInputs.hasOwnProperty(z)) {
320                     //console.log(z)

```



```

321         mainInputs[z].shape.set("text",op.
322             toString());
323         canvas.renderAll();
324     }
325 }
326 }
327 return (op);
328 }

```

### 2.3.4 UI Styling

```

1  /*@import url(http://fonts.googleapis.com/css?family=Source+
2     Sans+Pro:400,700);*/
3  * {
4     margin: 0;
5     padding: 0;
6     font-family: 'Source Sans Pro', Helvetica, Arial, sans-
7         serif;
8     cursor: default;
9     -webkit-user-select: none;
10    -moz-user-select: none;
11    -ms-user-select: none;
12    -o-user-select: none;
13    user-select: none;
14 }
15 body {
16     width: 100%;
17     height: 100%;
18     overflow: hidden;
19     background: #ddd;
20 }
21 header {
22     z-index: 10;
23 }
24 header .title {
25     background: #49c;
26     border-bottom: 1px solid #48b;
27     padding: 10px;
28     color: #fff;
29     font-size: 18px;
30 }
31 header .toolbar {
32     background: #efefef;
33     border-bottom: 1px solid #ccc;
34 }
35 header .toolbar ul {
36     display: block;

```

```

36 }
37 header .toolbar ul li {
38     display: inline-block;
39     padding: 10px;
40     cursor: pointer;
41     border-right:1px solid transparent;
42     border-left:1px solid transparent;
43 }
44 header .toolbar ul li:hover {
45     background: #ddd;
46     border-right:1px solid #ccc;
47     border-left:1px solid #ccc;
48 }
49 #workspace {
50     position: relative;
51     overflow: hidden;
52 }
53 /*canvas{
54     position: absolute;
55     background: #fff;
56     border: 1px solid #ccc;
57     box-shadow: 0 0 3px rgba(0,0,0,0);
58     transition:transform .2s ease-in-out;
59     z-index: 9;
60 }*/
61 .menu-selected{
62     background: #ddd!important;
63     border-right:1px solid #ccc!important;
64     border-left:1px solid #ccc!important;
65 }
66 #cClear{
67     float: right;
68 }

```

### 2.3.5 The Main HTML File

```

1 <!DOCTYPE html>
2 <html lang="en">
3
4 <head>
5     <meta charset="UTF-8">
6     <title>LogicSimJS</title>
7     <link rel="stylesheet" href="system/css/main.css">
8     <script src="system/js/fabric.min.js"></script>
9 </head>
10
11 <body>
12     <header>
13         <div class="title"><strong>LogicSimJS</strong>

```

```
14     </div>
15     <div class="toolbar">
16         <ul>
17             <li id="aCreate">AND</li>
18             <li id="oCreate">OR</li>
19             <li id="nCreate">NOT</li>
20             <li id="iCreate">Input</li>
21             <li id="lCreate">Line</li>
22             <li id="cClear">Clear</li>
23         </ul>
24     </div>
25 </header>
26 <div id="workspace">
27     <canvas id="canvas"></canvas>
28 </div>
29 <script src="system/js/main.js"></script>
30 <script src="system/js/moduleShapes.js"></script>
31 <script src="system/js/createShapes.js"></script>
32 </body>
33
34 </html>
```

### 3 Project Timeline

Title/Weeks	Week1	Week2	Week3	Week4	Week5	Week 6
Preparation/Proposal	■					
Coding		■	■	■	■	
Documentation				■	■	■
Fine Tuning/Submission						■

## References

- [1] “Active hdl.” <http://web.archive.org/web/20080207010024>. Accessed: December 17, 2014.
- [2] “Schemeit.” <http://www.digikey.com/schemeit>. Accessed: December 17, 2014.