//AND Gate

//AND\_array stores information for multiple AND gates

var AND\_array = [];

//Stores function and function parameters in a hidden text bar

var store\_function = document.getElementById('function\_selection');

function ANDgate(AND\_line, time)

{

var AND\_currentLine;

var AND\_logicOutput;

var AND\_findLL;

/\*For loop below searches AND\_array for the line called, then stores the name to

use later.

\*/

for (var AND\_lineFinder = 0; AND\_lineFinder < AND\_array.length; AND\_lineFinder++)

{

if (AND\_line == AND\_array[AND\_lineFinder].name)

{

AND\_currentLine = AND\_array[AND\_lineFinder];

break;

}

}

/\*For loop finds the logic\_line the name belongs to and stores the logic state

the line is currently in.

\*/

for (var LL\_index = 0; LL\_index < logic\_lines.length; LL\_index++)

{

if (logic\_lines[LL\_index].name == AND\_line)

{

AND\_findLL = 'logic\_lines[' + LL\_index + ']';

AND\_currentLine.prevlogic = eval(AND\_findLL).logic\_state;

break;

}

}

/\*For loop checks if each input line is HIGH If not, the logic state of the gate is

set to LOW and the for lop is immediately exited.

\*/

for (var AND\_inputIter = 0; AND\_inputIter < AND\_currentLine.inputs.length; AND\_inputIter++)

{

AND\_logicOutput = 'HIGH';

if (eval(AND\_currentLine.inputs[AND\_inputIter]).logic\_state == 'LOW')

{

AND\_logicOutput = 'LOW';

break;

}

}

/\*If statement checks if the logic state of the line changed. If so, the current time

is stored into the current line's inputs

\*/

if (AND\_logicOutput != AND\_currentLine.prevlogic)

{

//If statement checks if the gate has delay or has already stabilized

if (eval(AND\_findLL).gateDelay != 0 &&

eval(AND\_findLL).gate\_delayLevel != 0 &&

eval(AND\_findLL).gate\_delayLevel != logic\_vary)

{

/\*If statement checks if:

The signal is decreasing and the gate needs to change into a HIGH

The Signal is increasing and the gate needs to change into a LOW

If any of these parameters are true, the current time is added to the inputs

array of the gate

\*/

if((AND\_currentLine.prev\_gateInc > eval(AND\_findLL).gate\_delayLevel &&

AND\_logicOutput == 'HIGH') ||

(AND\_currentLine.prev\_gateInc < eval(AND\_findLL).gate\_delayLevel &&

AND\_logicOutput == 'LOW'))

{

eval(AND\_findLL).inputs.push(time);

order\_existingArray(LL\_index);

}

}

//Adds an input into the gate's inputs array if if statement isn't true

else

{

document.getElementById('test').innerHTML = 'Gate has an input!';

eval(AND\_findLL).inputs.push(time);

order\_existingArray(LL\_index);

}

AND\_currentLine.prev\_gateInc = eval(AND\_findLL).gate\_increment;

}

}

function ANDgate\_setup()

{

//A table is creates to display all the inputs available to the gate with a confirm button

var table\_inputs = "<table><tr><th>AND</th></tr>";

for (var table\_count = 0; table\_count < logic\_lines.length; table\_count++)

{

table\_inputs += "<tr><td class='inputTable'>" +

logic\_lines[table\_count].name +

"</tr></td>";

}

table\_inputs += "</table>" +

"<button id='createObject'>Finish Gate</button>" +

"<button id='cancelGate'>Cancel</button>";

document.getElementById('moreLogic').innerHTML = table\_inputs;

$(document).ready(function()

{

/\*When selecting gate inputs, clicking on text toggles class to indicate

selection and to find its location in logic\_lines[].

\*/

$(".inputTable").click(function()

{

$(this).toggleClass("selected\_input");

});

$("#createObject").click(function()

{

// IC\_nameLF grabs name from name text box

var IC\_nameLF = document.getElementById('create\_name').value;

//Gets the amount of names with 'selected\_input' class

var AND\_classCounter = document.getElementsByClassName('selected\_input');

//Used to store gate name and input location for gate

var AND\_inputObject = {name: IC\_nameLF, inputs:[], prevlogic: '', prev\_gateInc: 0};

/\*Two for loops are used to get the name for each table element with the selected\_input class

and finds the logic line the name matches with.

\*/

for (var iter\_classCounter = 0; iter\_classCounter < AND\_classCounter.length; iter\_classCounter++)

{

for (var LL\_index = 0; LL\_index < logic\_lines.length; LL\_index++)

{

/\*When the line is found, the if statement stores the name into the inputs and breaks for loop

for (hopefully) efficient processing.

\*/

if (logic\_lines[LL\_index].name == AND\_classCounter[iter\_classCounter].innerHTML)

{

AND\_inputObject.inputs.push('logic\_lines['+ LL\_index +']');

break;

}

}

}

//Stores all accumulated information into AND\_array

AND\_array.push(AND\_inputObject);

//Storing the function as a text prevents calling when creating the function

store\_function.value = 'ANDgate("' + IC\_nameLF + '", logic\_timing)';

document.getElementById('moreLogic').innerHTML = "";

});

$("#cancelGate").click(function()

{

document.getElementById('moreLogic').innerHTML = "";

})

});

}