//The array and two objects below are used to test program

/\*var test\_inputs = [18,10,50,33,60,117,301,228,290];

var new\_inputs = {

name:'CLK', //Name of logic line

location:logic\_vary+logic\_space, //Location of logic line on canvas

inverted:false, //Weather or not the signal is inverted

/\*Logic inputs. INPUTS MUST BE IN AN ARRAY!! Sort method

numerically sorts the input values

\*/

/\*inputs:test\_inputs.sort(function(a, b){return a - b}),

start: false,

input\_iter:0, //Used to keep track of the inputs

gateDelay:6, //How long the signal takes to change from HIGH to LOW

gate\_delayLevel:0, //Used to save logic location

prev\_logic:0, //Used to save previous logic state

gate\_signalChange:0, //Indicates if the logic goes LOW to HIGH of HIGH to LOW

gate\_increment: 0, //Used to calculate height of logic level with a gate delay

/\*key: [not, flip, change, function]

[draw\_logic] saves the first three variables to use later in

calculating the location of the previous and current logic.

\*/

/\*logic\_output:[0,0,0,

function ()

{

return this[0]-this[1]\*this[2];

}]

this.logic\_change = [get\_logicalHIGH.value, get\_logicalLOW.value]; //How high or low the logic needs to go before the logic state changes

this.logic\_state = "LOW"; //Stores the logic state

this.logic\_function = [0, null]; //Stores the logic function

};

var new\_inputs2 =

{

name:'7404',

location:(logic\_vary+logic\_space)\*2,

inverted:true,

gateDelay:0,

inputs: [45,28,80,56,77,140,350,200,266].sort(function(a, b){return a - b}),

start: false,

gate\_delayLevel:0,

gate\_increment:0,

prev\_logic:0,

gate\_signalChange:0,

input\_iter:0,

logic\_output:[0,0,0,

function ()

{

return this[0]-this[1]\*this[2];

}]

this.logic\_change = [get\_logicalHIGH.value, get\_logicalLOW.value];

this.logic\_state = "LOW";

this.logic\_function = [0, null];

};\*/