Garrett Barton

Section 2/Group 1

SEMESTER PROJECT

Overview statement

IMPLEMENT A TRAFFIC LIGHT SYSTEM AS DETAILED IN EXPERIMENT DESRITPTION.

Digital Input(s):

1. Data inputs: NS\_sensor, EW\_sensor,resen
2. Select lines: N/A
3. Clock:clk

Digital Output(s)/ Equations:

assign NS\_G = L[3];

assign NS\_Y = L[2];

assign NS\_R = L[1];

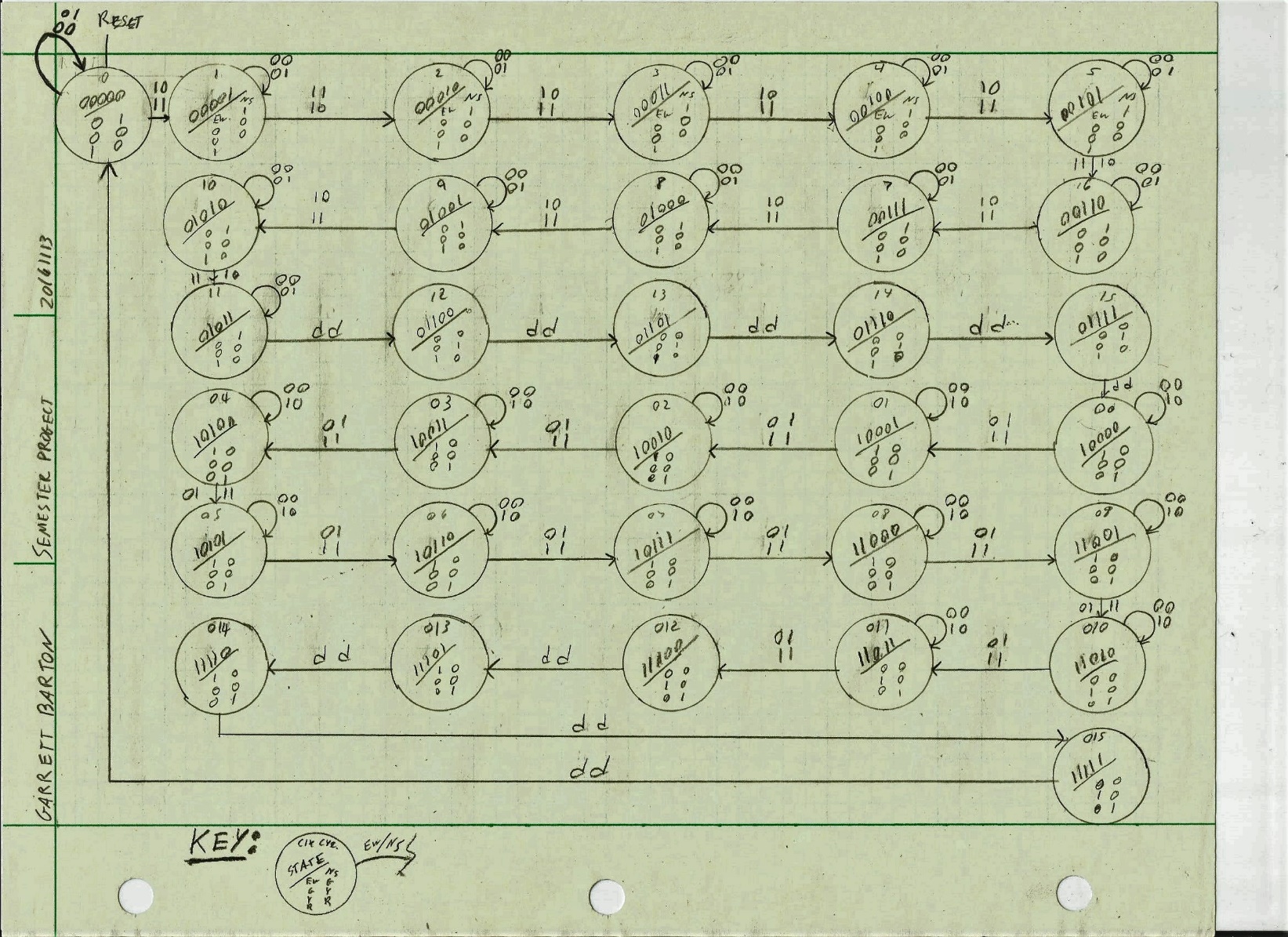
///////////////

assign EW\_G = L[6];

assign EW\_Y = L[5];

assign EW\_R = L[4];

Logic Diagram:



\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\

**MAIN CODE**

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

`timescale 1ns / 1ps

//////////////////////////////////////////////////////////////////////////////////

// Company: LSU,EE DEPT.

// Engineer: GARRETT BARTON

//

// Create Date: 21:43:45 11/13/2016

// Design Name: SEMESTER PROJECT FA16

// Module Name: BARTON\_CODE

// Project Name: BARTON

// Target Devices: BASYS 2

// Tool versions:

// Description: TRAFFIC LIGHT SYSTEM

//

// Dependencies:

//

// Revision:

// Revision 0.01 - File Created

// Additional Comments:

//

//////////////////////////////////////////////////////////////////////////////////

module BARTON\_CODE(

input NS\_sensor,

input EW\_sensor,

input clk,

input resetn,

output NS\_R,

output NS\_G,

output NS\_Y,

output EW\_R,

output EW\_G,

output EW\_Y

);

reg [5:1] t = 5'b00000;/////casing over states

reg [6:1] L = 6'b000000;///to hold states lights could be in

parameter [6:1] //EW G Y R // THEN NS G Y R////casing over lights

A = 6'b 001100, ///NS G///EW R

B = 6'b 001010, ///NS Y///EW R

C = 6'b 100001, ///NS R///EW G

D = 6'b 010001; ///NS R///EW Y

// Define the sequential block

always @(posedge clk)

if (resetn == 1) begin t <= 5'b00000; L <= A;end

else

case(t)

5'b00000: case({EW\_sensor,NS\_sensor})

2'b00,2'b01: if(1) begin t <= 5'b00000; L <= A; end

2'b10,2'b11: if(1) begin t <= 5'b00001; L <= A; end

endcase

5'b00001: case({EW\_sensor,NS\_sensor})

2'b00,2'b01: if(1) begin t <= 5'b00001; L <= A; end

2'b10,2'b11: if(1) begin t <= 5'b00010; L <= A; end

endcase

5'b00010: case({EW\_sensor,NS\_sensor})

2'b00,2'b01: if(1) begin t <= 5'b00010; L <= A; end

2'b10,2'b11: if(1) begin t <= 5'b00011; L <= A; end

endcase

5'b00011: case({EW\_sensor,NS\_sensor})

2'b00,2'b01: if(1) begin t <= 5'b00011; L <= A; end

2'b10,2'b11: if(1) begin t <= 5'b00100; L <= A; end

endcase

5'b00100: case({EW\_sensor,NS\_sensor})

2'b00,2'b01: if(1) begin t <= 5'b00100; L <= A; end

2'b10,2'b11: if(1) begin t <= 5'b00101; L <= A; end

endcase

5'b00101: case({EW\_sensor,NS\_sensor})

2'b00,2'b01: if(1) begin t <= 5'b00101; L <= A; end

2'b10,2'b11: if(1) begin t <= 5'b00110; L <= A; end

endcase

5'b00110: case({EW\_sensor,NS\_sensor})

2'b00,2'b01: if(1) begin t <= 5'b00110; L <= A; end

2'b10,2'b11: if(1) begin t <= 5'b00111; L <= A; end

endcase

5'b00111: case({EW\_sensor,NS\_sensor})

2'b00,2'b01: if(1) begin t <= 5'b00111; L <= A; end

2'b10,2'b11: if(1) begin t <= 5'b01000; L <= A; end

endcase

5'b01000: case({EW\_sensor,NS\_sensor})

2'b00,2'b01: if(1) begin t <= 5'b01000; L <= A; end

2'b10,2'b11: if(1) begin t <= 5'b01001; L <= A; end

endcase

5'b01001: case({EW\_sensor,NS\_sensor})

2'b00,2'b01: if(1) begin t <= 5'b01001; L <= A; end

2'b10,2'b11: if(1) begin t <= 5'b01010; L <= A; end

endcase

5'b01010: case({EW\_sensor,NS\_sensor})

2'b00,2'b01: if(1) begin t <= 5'b01010; L <= A; end

2'b10,2'b11: if(1) begin t <= 5'b01011; L <= A; end

endcase

5'b01011: case({EW\_sensor,NS\_sensor})/////11////////////////////////

2'b00,2'b01: if(1) begin t <= 5'b01011; L <= A; end

2'b10,2'b11: if(1) begin t <= 5'b01100; L <= B; end

endcase

5'b01100: case({EW\_sensor,NS\_sensor})/////12////////////////////////

2'b00,2'b01,2'b10,2'b11: if(1) begin t <= 5'b01101; L <= B; end

endcase

5'b01101: case({EW\_sensor,NS\_sensor})

2'b00,2'b01,2'b10,2'b11: if(1) begin t <= 5'b01110; L <= B; end

endcase

5'b01110: case({EW\_sensor,NS\_sensor})

2'b00,2'b01,2'b10,2'b11: if(1) begin t <= 5'b01111; L <= B; end

endcase

5'b01111: case({EW\_sensor,NS\_sensor})///15/////////////////////////

2'b00,2'b01,2'b10,2'b11: if(1) begin t <= 5'b10000; L <= C; end

endcase

5'b10000: case({EW\_sensor,NS\_sensor})///00/////////////////////////

2'b00,2'b10: if(1) begin t <= 5'b10000; L <= C; end

2'b01,2'b11: if(1) begin t <= 5'b10001; L <= C; end

endcase

5'b10001: case({EW\_sensor,NS\_sensor})

2'b00,2'b10: if(1) begin t <= 5'b10001; L <= C; end

2'b01,2'b11: if(1) begin t <= 5'b10010; L <= C; end

endcase

5'b10010: case({EW\_sensor,NS\_sensor})

2'b00,2'b10: if(1) begin t <= 5'b10010; L <= C; end

2'b01,2'b11: if(1) begin t <= 5'b10011; L <= C; end

endcase

5'b10011: case({EW\_sensor,NS\_sensor})

2'b00,2'b10: if(1) begin t <= 5'b10011; L <= C; end

2'b01,2'b11: if(1) begin t <= 5'b10100; L <= C; end

endcase

5'b10100: case({EW\_sensor,NS\_sensor})

2'b00,2'b10: if(1) begin t <= 5'b10100; L <= C; end

2'b01,2'b11: if(1) begin t <= 5'b10101; L <= C; end

endcase

5'b10101: case({EW\_sensor,NS\_sensor})

2'b00,2'b10: if(1) begin t <= 5'b10101; L <= C; end

2'b01,2'b11: if(1) begin t <= 5'b10110; L <= C; end

endcase

5'b10110: case({EW\_sensor,NS\_sensor})

2'b00,2'b10: if(1) begin t <= 5'b10110; L <= C; end

2'b01,2'b11: if(1) begin t <= 5'b10111; L <= C; end

endcase

5'b10111: case({EW\_sensor,NS\_sensor})

2'b00,2'b10: if(1) begin t <= 5'b10111; L <= C; end

2'b01,2'b11: if(1) begin t <= 5'b11000; L <= C; end

endcase

5'b11000: case({EW\_sensor,NS\_sensor})

2'b00,2'b10: if(1) begin t <= 5'b11000; L <= C; end

2'b01,2'b11: if(1) begin t <= 5'b11001; L <= C; end

endcase

5'b11001: case({EW\_sensor,NS\_sensor})

2'b00,2'b10: if(1) begin t <= 5'b11001; L <= C; end

2'b01,2'b11: if(1) begin t <= 5'b11010; L <= C; end

endcase

5'b11010: case({EW\_sensor,NS\_sensor})

2'b00,2'b10: if(1) begin t <= 5'b11010; L <= C; end

2'b01,2'b11: if(1) begin t <= 5'b11011; L <= C; end

endcase

5'b11011: case({EW\_sensor,NS\_sensor})////////011//////////////////

2'b00,2'b10: if(1) begin t <= 5'b11011; L <= C; end

2'b01,2'b11: if(1) begin t <= 5'b11100; L <= D; end

endcase

5'b11100: case({EW\_sensor,NS\_sensor})////////012/////////////////

2'b00,2'b01,2'b10,2'b11: if(1) begin t <= 5'b11101; L <= D; end

endcase

5'b11101: case({EW\_sensor,NS\_sensor})

2'b00,2'b01,2'b10,2'b11: if(1) begin t <= 5'b11110; L <= D; end

endcase

5'b11110: case({EW\_sensor,NS\_sensor})

2'b00,2'b01,2'b10,2'b11: if(1) begin t <= 5'b11111; L <= D; end

endcase

5'b11111: case({EW\_sensor,NS\_sensor})

2'b00,2'b01,2'b10,2'b11: if(1) begin t <= 5'b00000; L <= D; end

endcase

endcase

///////////////////////////////assignments//////////////////////////////

assign NS\_G = L[3];

assign NS\_Y = L[2];

assign NS\_R = L[1];

///////////////

assign EW\_G = L[6];

assign EW\_Y = L[5];

assign EW\_R = L[4];

endmodule

\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\

**TEST BENCH**

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

`timescale 1ns / 1ps

////////////////////////////////////////////////////////////////////////////////

// Company:

// Engineer:

//

// Create Date: 21:44:31 11/13/2016

// Design Name: BARTON\_CODE

// Module Name: F:/2016-2017/FA 16/DIG LAB 2731/SEMESTER PROJECT/BARTON\_CODE\_TB.v

// Project Name: BARTON

// Target Device:

// Tool versions:

// Description:

//

// Verilog Test Fixture created by ISE for module: BARTON\_CODE

//

// Dependencies:

//

// Revision:

// Revision 0.01 - File Created

// Additional Comments:

//

////////////////////////////////////////////////////////////////////////////////

module BARTON\_CODE\_TB;

// Inputs

reg NS\_sensor;

reg EW\_sensor;

reg clk;

reg resetn;

// Outputs

wire NS\_G;

wire NS\_Y;

wire NS\_R;

wire EW\_G;

wire EW\_Y;

wire EW\_R;

// Instantiate the Unit Under Test (UUT)

BARTON\_CODE uut (

.NS\_sensor(NS\_sensor),

.EW\_sensor(EW\_sensor),

.clk(clk),

.resetn(resetn),

.NS\_G(NS\_G),

.NS\_Y(NS\_Y),

.NS\_R(NS\_R),

.EW\_G(EW\_G),

.EW\_Y(EW\_Y),

.EW\_R(EW\_R)

);

initial begin

// Initialize Inputs

NS\_sensor = 0;

EW\_sensor = 1;

clk = 0;

resetn = 0;

// Wait 100 ns for global reset to finish

#100;

// Add stimulus here

end

//////////////////////////////////////////////////

//TESTING CODE BELOW

always

begin

#15 clk = ~clk;

end

////////////////////////////////////////////////////////////////

/\*always@(posedge clk)

#15

begin

#150

NS\_sensor = 0;

EW\_sensor = 0;

#45

NS\_sensor = 0;

EW\_sensor = 1;

end

\*/

endmodule

\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\

**UCF FILE**

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

NET "NS\_sensor" LOC = "B4"; //SW3

NET "EW\_sensor" LOC = "P11"; //SW0

NET "clk" LOC = "B2"; //PUSH BUTTON 1 ON PLUG-IN

NET "resetn" LOC = "N3";

//LED 5-7

NET "NS\_G" LOC = "G1"; //LED7

NET "NS\_Y" LOC = "P4";//LED6

NET "NS\_R" LOC = "N4";//LED5

//LED 0-2

NET "EW\_G" LOC = "P7";//LED2

NET "EW\_Y" LOC = "M11";//LED1

NET "EW\_R" LOC = "M5";//LED0

//TO USE PUSH BUTTON PLUG-IN

NET "clk" CLOCK\_DEDICATED\_ROUTE = FALSE;