**Traffic Signal Controller**

Specifications:

* The traffic signal for the main highway gets the highest priority because cars are continuously present on the main highway. Thus, the main highway signal remains green by default.
* Occasionally cars from the country road arrive at the traffic signal. The traffic signal for country road must turn green only long enough to let the cars on the country road go.

A cross with text on it

Description automatically generated

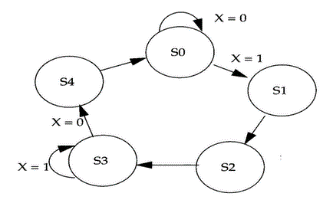
* As soon as there are no cars on the country road, the country road traffic signal turns yellow and then red, and the traffic signal on the main highway turns green again.
* There is a sensor to detect cars waiting on the country road. The sensor sends a signal X as input to the controller. X=1 if there are cars on the country road, otherwise x=0
* There are delays on transition from S1 to S2, from S2 to S3 and from S4 to S0. The delays must be controllable.

Problem: Build a Verilog model for traffic signal control using the state machine

State table

|  |  |
| --- | --- |
| **State** | **Signals** |
| S0 | Hwy = Green, Cntry = Red |
| S1 | Hwy = Yellow, Cntry = Red |
| S2 | Hwy = Red, Cntry = Red |
| S3 | Hwy = Red, Cntry = Green |
| S4 | Hwy = Red, Cntry = Yellow |

State diagram



Verilog HDL Code:

`timescale 1ns / 1ps

`define TRUE 1'b1 //compiler directive-a statement that causes the compiler to take a specific action during compilation

`define FALSE 1'b0

module main(clk, clear, X, hwy, cntry);

input clk, clear;

input X; //if true indicates that there is car in the country road, otherwise false

output reg [1:0] hwy, cntry; //2bit output for 3 states(red, green, yellow)

//internal state variables

reg [2:0] state;

reg [2:0] next\_state;

//states declaration as parameter -used to pass a constant to the module when it is instantiated

parameter S0=3'd0;

parameter S1 = 3'd1;

parameter S2 = 3'd2;

parameter S3 = 3'd3;

parameter S4 = 3'd4;

parameter RED = 2'd0,

YELLOW = 2'd1,

GREEN = 2'd2;

//delays added

parameter Y2R\_delay = 3; //3 delay unit from yellow to red

parameter R2G\_delay = 2; //2 delay unit from red to green

//state change only at posege of clk

always @(posedge clk)

if(clear)

state<=S0;

else

state <= next\_state;

always @(state)

begin

hwy = GREEN; //0 //default values priority given to highway

cntry = RED; //2

case(state) //state defination

S0: ;

S1: hwy = YELLOW;

S2: hwy = RED;

S3: begin

hwy = RED;

cntry = GREEN;

end

S4: begin

hwy = RED;

cntry = YELLOW;

end

endcase

end

//logic - what happens in each state

always@(state or X)

begin

case(state)

S0: if(X)

next\_state = S1;

else

next\_state = S0;

S1: begin //delaying

repeat (Y2R\_delay)@(posedge clk) next\_state=S1;

next\_state = S2;

end

S2: begin //delaying

repeat (R2G\_delay)@(posedge clk)next\_state=S2;

next\_state = S3;

end

S3: if(X)

next\_state = S3;//same state if car present

else

next\_state = S4;//next state if no car

S4: begin //delaying

repeat (Y2R\_delay)@(posedge clk)next\_state=S4;

next\_state = S0;

end

default: next\_state = S0;

endcase

end

endmodule

Verilog HDL testbench:

`timescale 1ns / 1ps

`define TRUE 1'b1 //compiler directive-a statement that causes the compiler to take a specific action during compilation

`define FALSE 1'b0

module testbench;

reg clk, clear;

reg X; //if true indicates that there is car in the country road, otherwise false

wire [1:0] hwy, cntry;

main uut(clk, clear, X, hwy, cntry);

initial

$monitor($time,"highway = %b, country = %b, X=%b",hwy, cntry, X);

//set clock

initial begin

clk = `FALSE; //clk=0

forever #5 clk = ~clk;

end

initial

begin

clear = `TRUE; //clear=1, reset state to s0

repeat (5)@(negedge clk); //instead of writing delay as #5, it can be written like this

clear = `FALSE; //clear=0

X=`FALSE; //X=0

#100 X=`TRUE;

#150 X=`FALSE;

#100 X=`TRUE;

#150 X=`FALSE;

clear = `TRUE; //clear=1, reset state to s0

repeat (5)@(negedge clk); //instead of writing delay as #5, it can be written like this

clear = `FALSE; //clear=0

#100 X=`TRUE;

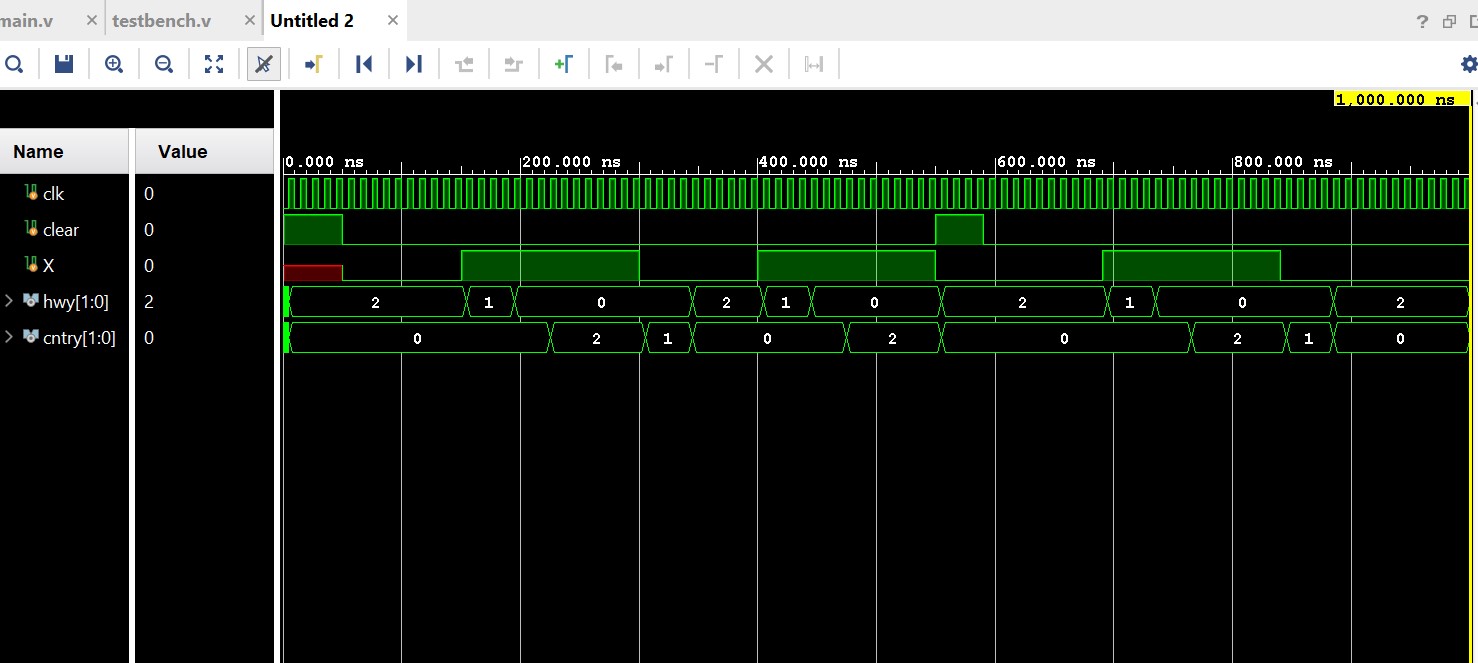
#150 X=`FALSE;

#100; $stop;

end

endmodule

Waveform



Console

A table of numbers and symbols

Description automatically generated

References:

* Verilog HDL A Guide to Digital Design and Synthesis by Samir Palnitkar