CS203 PROJECT REPORT:
TRAFFIC CONTROL SYSTEM

Team members-

INAYAT KAUR (2020CSB1088)

ISHA GOYAL (2020CSB1089)

TARUSHI (2020CSB1135)

Professor's name

NEERAJ GOEL

INTRODUCTION

The project deals with the implementation of a traffic signal at 4-way crossing. There are a total of 4 signals having red, green and yellow LEDs each. In the project demonstration we consider the red light of a signal to be ON for 10 sec, while the green and yellow lights remain ON for a total of 5 secs each.

The timer corresponding to each signal is displayed. Unlike how a normal traffic light timer displays the amount of time left before the signal turns green or red, in our project, we display the time left until the light which is currently switched ON, let it be green, yellow or red, remains ON.

Along with this, 4 LEDs are used to indicate when a signal has been broken. With the help of a switch, we indicate that a vehicle has crossed the signal and if the signal at that time is red, then the corresponding LED switches ON and a buzzer starts ringing. This indicates that a traffic light has been jumped.

All of the above has been implemented on a hardware called FPGA, using Verilog.

FPGA

Field Programmable Gate Array, FPGA, is an integrated circuit that can be programmed by the user for a specific use. FPGAs consist of logical modules connected by routing channels. Each module is made up of a programmable lookup table that controls the elements that each cell consists of and performs logical functions of the elements that make up the cell. In addition to the lookup table, each cell contains cascaded adders enabling addition to be done. Subtraction can also be done by changing the logical states of the input. Beyond these, there are also registers and multiplexers .

BLOCK DIAGRAM

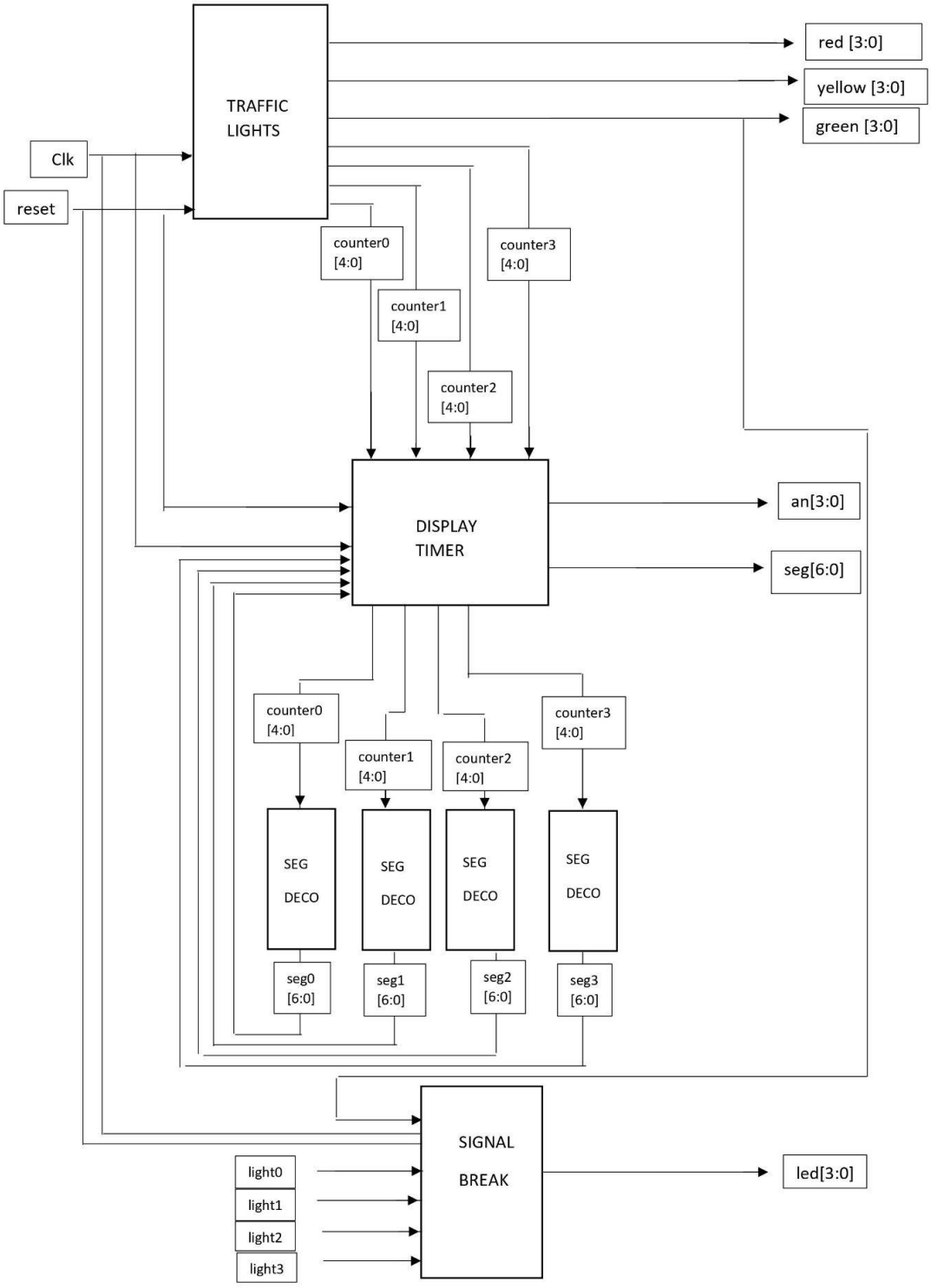

FIG1-BLOCK DIAGRAM

MODULES

Module is a block of verilog code that implements a certain functionality and in our project we have used 4 modules -

1. timer_display- This is the module for displaying time.

FPGA has an in-built 7 segment display and among the corresponding segments of each display (left,center-left,center-right,right) we can choose which one to keep active at one time. We then flip the displays at such a rate that the human eye perceives that each display is showing a different number. In our project, the displays flip every 0.001 second.

2.signal_break- This module is used when a signal gets broken.

In this module, we take the 4-bit green signal, generated as an output from the traffic_lights module, as an input. Along with this, it takes input from four switches of the FPGA. Each switch corresponds to one LED and one bit of the green signal. If the switch is toggled ON, and the corresponding green signal bit is 0, then the LED switches ON and the buzzer starts ringing.

3.segment_decode- This is the module used for displaying decimal numerals on a 7 segment display.

This module converts a decimal digit to its corresponding 7-segment code.
The project uses 4 segment_decode blocks to convert four 4-bit counters (counter0, counter1, counter2 and counter3) to 7-segment codes, which it outputs.It has been instantiated in timer_display.

4.traffic_light - This is the main module used for the functionality of the traffic light.

This module controls the working of the LEDs of the 4 traffic lights.
The first signal is initialized to green, the second to yellow and the other two to red. It has a counter, which keeps increasing by 1 every second. Every 5 seconds (counted with the help of the counter), the state of the traffic lights changes. Green turns to yellow, yellow to red every 5 seconds, while red turns to green every 10 seconds.
The in-built clock of the FPGA has a frequency of 100MHz. This clock needs to be slowed down to 1Hz and hence we use the lights_clk signal.

The traffic lights module consists of two state machines, one of them being that of the counter. The counter state machine has a total of 20 states. We start with the initial state S0. After every second, the counter enters a new state. The output of the state machine becomes 1 after the 4th, 9th, 14th and 19th state, as it is after these states that the changes in traffic lights occur.

The other state machine is of the traffic lights having four states. Initially we are in state S0. After every 5 states of the counter we move to the next state of the traffic lights. So that after 20 states of the counter we get back to the initial state S0.

Present state	Traffic light 1	Traffic light 2	Traffic light 3	Traffic light 4	Next State
S0	Green	Yellow	Red	Red	S1
S1	Red	Green	Yellow	Red	S2
S2	Red	Red	Green	Yellow	S3
S3	Yellow	Red	Red	Green	S0

The two state machines are interlinked with one another. The output received from the counter state machine is used in the traffic lights state machine.

FPGA PINS USED

1. **Switch 16 (connected to reset) :**

   Reset is used to set things back to the original state. If reset = 1, the timers, counters and LEDs return to the initial state.

2. **Switches 15-12 (connected to light 0-3) :**

   If a switch is toggled ON, it indicates that a vehicle has moved across the corresponding traffic light, i.e, if light0 is toggled ON, then a vehicle has moved across the first traffic light. If the switch gets toggled when the corresponding traffic light is not green, then, the traffic light is considered to be broken and an LED lights up.

3. **LEDs 16-5 (connected to the traffic lights) :**

   The 12 LEDs represent the traffic light in a sequential order, where the first 3 represent red[0], yellow[0], green[0] and so on.

4. **LEDs 4-1 (connected to led) :**

   Each light corresponds to a particular traffic light. If the red light is jumped then the corresponding LED switches ON. For example, if the first traffic light is broken, then light0 switches ON.

5. **Display pins (connected to seg and an) :**

   We used the an and seg pins to display the the timer on the 7 segment display. ‘an’ allows us to choose the display (left, center-left, center-right, right) and seg allows us to choose the corresponding segment pins that need to be displayed.

6. **Buzzer Pin(L17 and R18):**

   When the signal is broken, the buzzer starts ringing.

DISPLAY PINS:	INPUT PINS:	OUTPUT PINS:
seg[0]=W7 seg[1]=W6 seg[2]=U8 seg[3]=V8 seg[4]=U5 seg[5]=V5 seg[6]=U7 an[0]=U2 an[1]=U4 an[2]=V4 an[3]=W4	clk=W5 reset=R2 light0=T1 light1=UI light2=W2 light3=R3	red[0]=L1 red[1]=P3 red[2]=V3 red[3]=U14 yellow[0]=P1 yellow[1]=U3 yellow[2]=V13 yellow[3]=U15 green[0]=N3 green[1]=W3 green[2]=V14 green[3]=W18 led[0]=V19 led[1]=U19 led[2]=E19 led[3]=U16 buzzer=L17, R18

FIG2 -FPGA

RESULT

Upon connecting the FPGA board and setting the reset switch to zero state, the 4 display segments, going from left to right, are assigned one traffic light each. They display the time left for an LED of a traffic light (which is currently ON) to switch OFF. LED'S 16-14 represent the first traffic light and so on. The initial states of the traffic lights are set as follows:

The first traffic light is green, with 5 seconds remaining till it turns red.
The second traffic light is yellow, with 5 seconds remaining till it turns green.
The third traffic light is red, with 5 seconds remaining till it turns yellow.
The fourth traffic light is red, with 10 seconds remaining till it turns yellow.

Any among the switches 15-12 can be set to one state and at that time if the green light for the corresponding switch is not ON then the buzzer starts ringing and corresponding LED(among LED'S 4-1) also gets turned ON, keeping in mind that the reset state is zero at this time. Switch 15 corresponds to the first Traffic signal and so on. LED 4 corresponds to the first Traffic signal and so on.

If the reset switch is set to 1, then all the LEDs as well as the display segments are reset to their initial states.

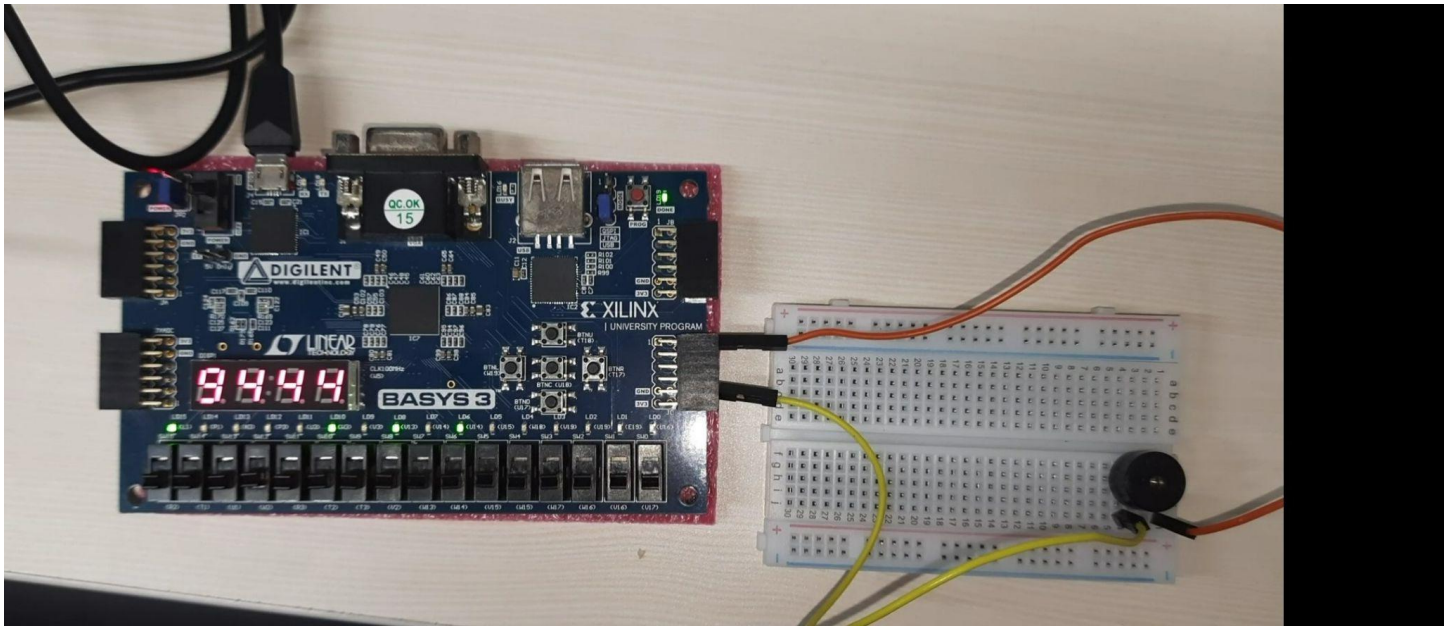

FIG 3- TRAFFIC CONTROL SYSTEM ON FPGA