

# **ATTENDANCE TRACKER**

## **Mini Project Report**

Submitted in partial fulfilment of the requirements for the degree of

BACHELOR OF TECHNOLOGY in

ELECTRONICS AND COMMUNICATION ENGINEERING

By

**SHAILAJA S GIRNI (221EC251)**

&

**MORA AKHIL TEJA (221EC133)**



DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING,  
NATIONAL INSTITUTE OF TECHNOLOGY KARNATAKA,  
SURATHKAL, MANGALORE -575025

APRIL 2024



# National Institute of Technology

## Karnataka

### CERTIFICATE

This is to certify that the thesis entitled, "ATTENDANCE TRACKER" Using counters and Simulation using VERILOG " submitted by SHAILAJA S GIRNI and MORA AKHIL TEJA in partial fulfillments for the requirements for the award of Bachelor of Technology Degree in Electronics & Communication Engineering at National Institute of Technology, Karnataka ,is an authentic work carried out by him under my supervision and guidance.

To the best of my knowledge, the matter embodied in the thesis has not been submitted to any other University / Institute for the award of any Degree or Diploma.

Prof. NIKHIL KS  
Dept. Electronics & Communication Engineering ,  
National Institute of Technology  
Karnataka, Surathkal - 575025

# **ACKNOWLEDGEMENT**

We would like to articulate our deep gratitude to our project guide Prof. Nikhil ks who has always been our motivation for carrying out the project. It is our pleasure to refer Microsoft Word exclusive of which the compilation of this report would have been impossible. An assemblage of this nature could never have been attempted without reference to and inspiration from the works of others whose details are mentioned in reference section. We acknowledge our indebtedness to all of them. Last but not the least, our sincere thanks to all of our friends who have patiently extended all sorts of help for accomplishing this undertaking.

SHAILAJA S GIRNI

MORA AKHIL TEJA

# ATTENDANCE TRACKER

The Objective of the project is to develop a digital attendance checker which updates the students about their current attendance status. The project consists the Verilog design which can either be used as a software model or fabricated into an actual circuit along with physical circuital model. The core objective is to keep a record of an individual's attendance and the total working days by which the system compares the attendance of an individual with a predetermined threshold and display him safe or unsafe message accordingly.

As in institutions attendance is also an integral part of assessment nowadays. It allows educators to identify students who may be struggling or falling behind, enabling timely intervention and support.

Even in corporations, schools and other functional bodies attendance plays a vital role in keeping record of participation, proper functioning of infrastructure, Evaluation of programs, Promotions, Safety and security etc. The mandatory actions can also be taken within time by keeping a good and manageable record of attendees.

The project consists flip- flop based Counters, Shift Registers, 4 Bit Adders, 4 Bit Comparators and a 7-Segment Display.

All the components work together to count and perform suitable operations on the recorded attendance. The current model is designed for maximum attendance of up to 24 sessions. The safe/unsafe status is given to students such that if there attendance is above 75 percent of the total working days then there are considered safe otherwise unsafe.

## PROBLEM STATEMENT:

The Attendance register is a device with everything related to his attendance criterion that a student *needs*. This includes the current attendance of the student, the total number of days the classes have been going on, the numbers of leaves the student can take or the number of days the student has to attend the class, if the attendance percentage of the student is safe, i.e., above 75%, and it even shows him if his attendance is short thereby giving him a FA grade.

## COUNTING THE ATTENDANCE:

To keep the record of attendance the present days and total working days are being counted simultaneously with the use of two different up-down counters.

### UP COUNTERS:

The Basic structure of up counters is made of sequentially arranged flip-flops with their output signals performing as input clock signals for the next arranged flip-flop. This means that each flip-flop will toggle on every clock pulse, but only if the previous flip-flop is in the "high" or "1" state. The above definition is mostly used for **Synchronous** **Counters**.

## COMPARISON WITH THE THRESHOLD:

To compare with the threshold, we need to perform some mathematical operations with the help of Shift Registers and 4-Bit Adders. After That the comparison will take place with the help of 4-Bit Comparators.

### Shift Register:

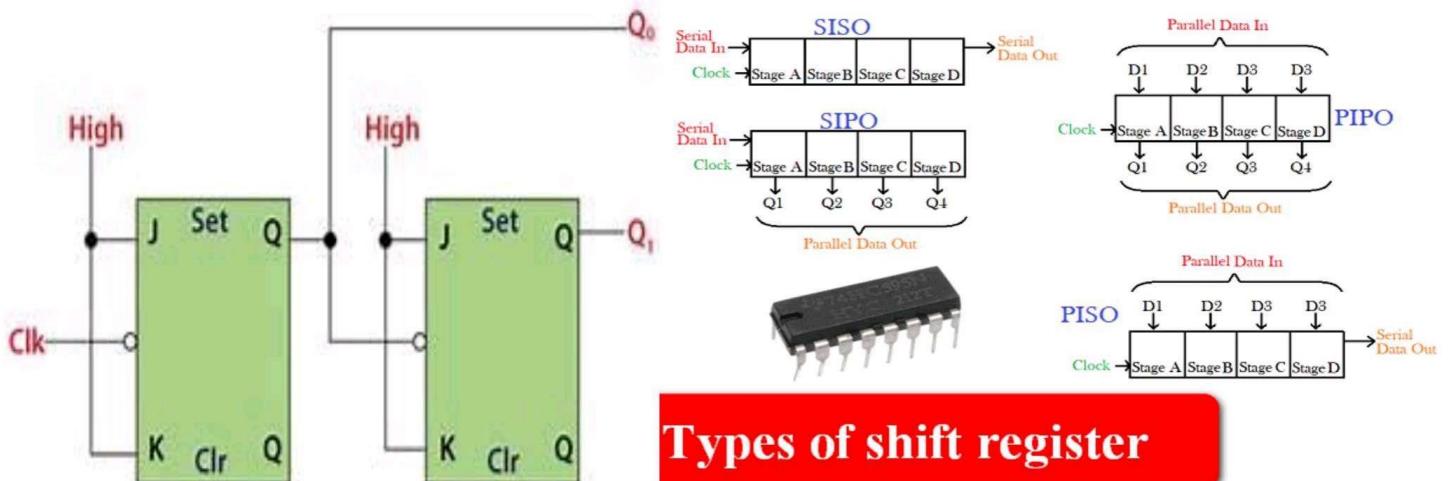
It is a digital sequential circuit consisting flip-flops arranged in a sequence each of which stores one bit. They can shift data bit by bit and have vast uses. They can also be used to perform some specific mathematical operations on stored binary data.

### 4-Bit Adders:

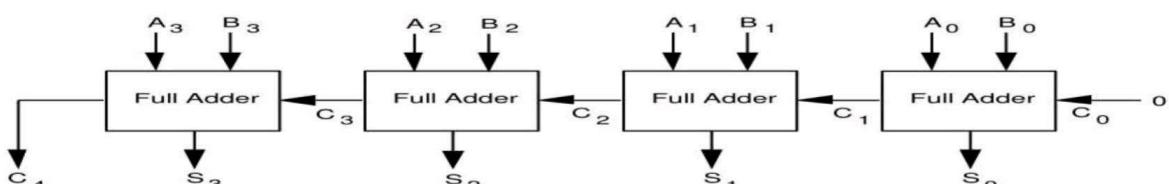
4-Bit Adders are used to add two 4-bit binary numbers. They are used as basic elements in many digital circuits. The 4-Bit Adders consists four full adders which adds two binary digits to and produces sum and carry. They are easy to design and can be managed according to the size of numbers to be added.

### 4-Bit Comparators:

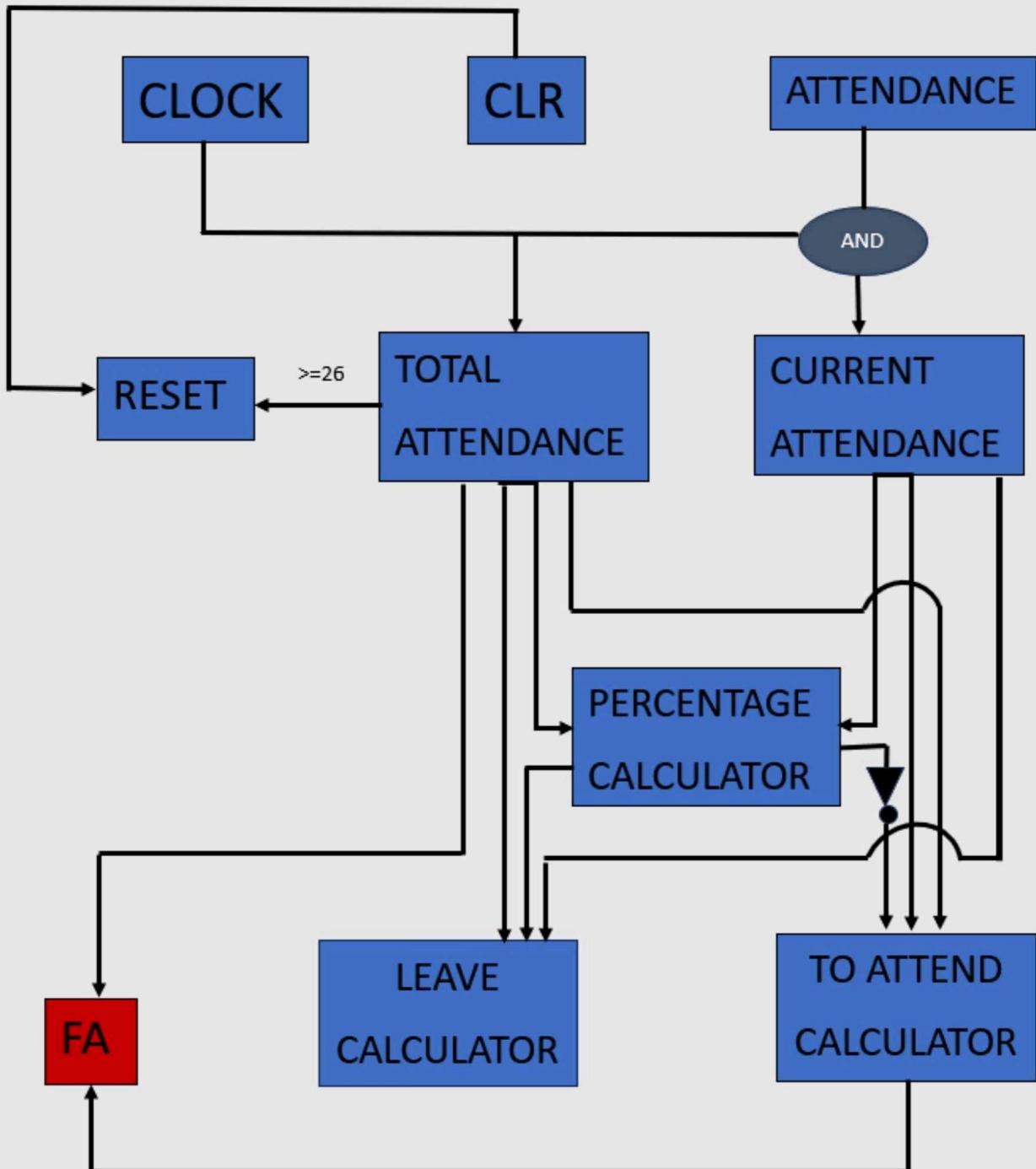
4-Bit Comparators are used to compare two 4-bit binary numbers. They consist four 1 bit comparators which compares two binary digits to check the cases of greater than, less than or equal to. The 1-bit comparators are connected in a parallel fashion, so that the corresponding bits of the two input numbers are compared simultaneously.



## 4-Bit Adder



<b>C</b>	1	1	1	0
<b>O : A</b>	0	1	1	0
<b>O : B</b>	0	0	1	1
<b>C4 : S</b>	1	0	1	0



## ARDUINO CODE

Given below is the MODULE code of the attendance register:

```
'timescale 1ns / 1ps
///////////////////////////////
// TEAMATES: SHAILAJA S GIRNI      & MORA AKHIL TEJA
// ROLL NUMBERS:221EC251      &    221EC133
//
// Create Date: 13.04.2024  14:27:00
// Module Name: attendance_tracker
// Project Name: ATTENDANCE TRACKER
// Tool Versions: 1.0
// Description: A USE FRIENDLY ATTENDANCE REGISTER FOR STUDENTS
//
// Revision 0.01 - File Created
// Additional Comments: MINI PROJECT FOR $ SEMESTER DSD  EC281
/////////////////////////////
module attendance_register (
    Input
    clock,
    input
    clear,
    input attendance,
    output reg is_
    safe,output reg
    FA,
    output reg [3:0] leaves,
    output reg [6:0]to_
    attend,
    output reg [6:0] total_attendance,
    output reg [6:0] current_
    attendance
);
```

```
//COUNTERS USED FOR BOTH TOTAL_ATTENDANCE AND CURRENT_ATTENDANCE
```

```
always @ (posedge clock) begin total_
attendance = total_attendance + 1;
if(attendance)
    current_attendance = current_attendance +
1;end
```

```
//RESET
```

```
always @ (*) begin
if (total_attendance >= 26 || clear==1) begin
    total_attendance = 1; current_attendance = 1 ; count1 = 1 ; count2 = 1 ; leaves = 0;
    to_attend = 0 ; is_safe = 0 ; FA = 0 ; temp_leaves = 0 ;
end
end
```

```
//COMPARATOR FOR ATTENDANCE SAFETY AND FA CALCULATOR
```

```
always @(posedge clock) begin
if((total_attendance + to_attend) >= 26
)fa = 1;
FA=fa;
count2 = current_attendance << 2;
count1 = (total_attendance << 1) + total_
attendance;if(count2 >= count1)
    is_safe = 1
;else
    is_safe = 0
;end
```

```
//TOTAL NUMBER OF LEAVES CALCULATOR
```

```
always @(posedge
clock)beginif(is_safe)begin
to_attend = 0;
if((current_attendance<<2) >= ((total_attendance<<1) + total_attendance))begin
```

```

temp_leaves = (current_attendance<<2) - ((total_attendance<<1) + total_
attendance); leaves = temp_leaves/3;
end
else
leaves = 0;
end

//TOTAL NUMBER OF DAYS TO ATTEND CLASSES TO BE SAFE CALCULATOR

else begin
leaves = 0;
if((total_attendance<<1) + total_attendance) >= (current_attendance<<2))
to_attend = ((total_attendance<<1) + total_attendance) - (current_attendance<<2);
else
to_
attend=0;end
end

```

endmodule

**Given below is the TESTBENCH for the given module:**

```

`timescale 1ns / 1ps
///////////////////////////////
// TEAMATES: SHAILAJA S GIRNI    &    MORA AKHIL TEJA
// ROLL NUMBERS:221EC251      &    221EC133
//
// Create Date: 13.04.2023 14:27:00
// Module Name: attendance_register
// Project Name: ATTENDANCE TRACKER
// Tool Versions: 1.0
// Description: A USER FRIENDLY ATTENDANCE REGISTER FOR STUDENTS
//
//
// Revision 0.01 - File Created
// Additional Comments: MINI PROJECT FOR 4 SEMESTER DSD: EC204
/////////////////////////////

```

```

module tb_attendance_register;
// Inputs
reg
clock;
reg attendance;
reg clear;

// Outputs
wire is_safe
;wire FA;
wire [3:0] leaves;
wire [6:0]to_attend;
wire [6:0] total_attendance;
wire [6:0] current_attendance;

// Attendance register instance
attendance_register uut_attendance_
checker(clock,
clear,
attendance,
is_safe,
FA,
leaves,
to_
attend,
total_attendance,
current_attendance
);
initial begin
clock <= 1'b0;
attendance=1; clear <= 1'b1;#1 attendance<=1;clr <= 1'b0;#11attendance<=0;clr <= 1'b0;#7

```

```

attendance<=1; clear <=
1'b0; #6attendance=0;
end

```

// Clock generator

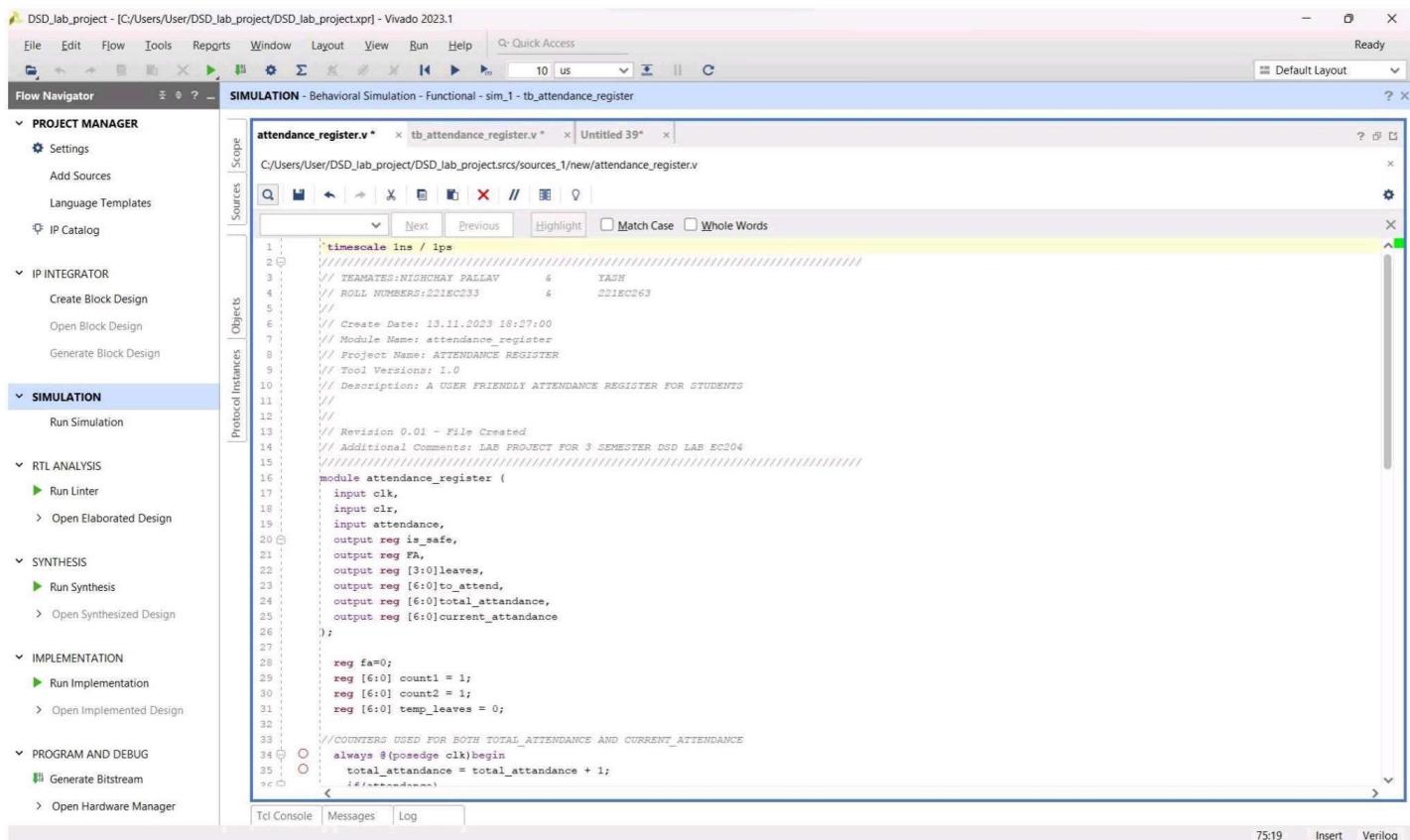
```

always #0.5 clock <=
~clock;

```

endmodule

**Given below are the snapshots of the above code:**



The screenshot shows the Vivado 2023.1 interface with the 'SIMULATION' tab selected. The central window displays the Verilog code for the `tb_attendance_register.v` testbench. The code includes comments at the top, a module definition for `attendance_register`, and an always block for the clock generator. The Vivado interface includes a project manager on the left, a search bar at the top, and various simulation and analysis tabs.

```

1: //timescale 1ns / 1ps
2: ///////////////////////////////////////////////////
3: /// TEA NAMES: NISHCHAY PALLAV      &          YASH
4: /// ROLL NUMBERS: 221EC233           &          221EC263
5: ///
6: /// Create Date: 13.11.2023 18:27:00
7: /// Module Name: attendance_register
8: /// Project Name: ATTENDANCE REGISTER
9: /// Tool Versions: 1.0
10: // Description: A USER FRIENDLY ATTENDANCE REGISTER FOR STUDENTS
11: ///
12: ///
13: // Revision 0.01 - File Created
14: // Additional Comments: LAB PROJECT FOR 3 SEMESTER OSD LAB EC204
15: ///////////////////////////////////////////////////
16: module attendance_register (
17:     input clk,
18:     input clr,
19:     input attendance,
20:     output reg is_safe,
21:     output reg FA,
22:     output reg [3:0] leaves,
23:     output reg [6:0] to_attend,
24:     output reg [6:0] total_attendance,
25:     output reg [6:0] current_attendance
26: );
27:
28:     reg fa=0;
29:     reg [6:0] count1 = 1;
30:     reg [6:0] count2 = 1;
31:     reg [6:0] temp_leaves = 0;
32:
33: //COUNTERS USED FOR BOTH TOTAL_ATTENDANCE AND CURRENT_ATTENDANCE
34: always @(posedge clk)begin
35:     total_attendance = total_attendance + 1;
36: end

```

**THE MODULES:**

DSD\_lab\_project - [C:/Users/User/DS... - Vivado 2023.1

File Edit Flow Tools Reports Window Layout View Run Help Q: Quick Access

Ready Default Layout

**Flow Navigator**

- PROJECT MANAGER
  - Settings
  - Add Sources
  - Language Templates
  - IP Catalog
- IP INTEGRATOR
  - Create Block Design
  - Open Block Design
  - Generate Block Design
- SIMULATION**
  - Run Simulation
- RTL ANALYSIS
  - Run Linter
  - Open Elaborated Design
- SYNTHESIS
  - Run Synthesis
  - Open Synthesized Design
- IMPLEMENTATION
  - Run Implementation
  - Open Implemented Design
- PROGRAM AND DEBUG
  - Generate Bitstream
  - Open Hardware Manager

**SIMULATION - Behavioral Simulation - Functional - sim\_1 - tb\_attendance\_register**

attendance\_register.v \* tb\_attendance\_register.v \* Untitled 39\* x

C:/Users/User/DS.../srcs\_1/new/attendance\_register.v

```

32 : //COUNTERS USED FOR BOTH TOTAL_ATTENDANCE AND CURRENT_ATTENDANCE
33 : always @(posedge clk)begin
34 :   total_attendance = total_attendance + 1;
35 :   if(attendance)
36 :     current_attendance = current_attendance + 1;
37 : end
38 :
39 :
40 :
41 : //RESET
42 : always @(*)begin
43 :   if(total_attendance >= 26 || clr==1 ) begin
44 :     total_attendance = 1; current_attendance = 1 ; count1 = 1 ; count2 = 1 ; leaves = 0 ;
45 :     to_attend = 0 ; is_safe = 0 ; FA = 0 ; temp_leaves = 0 ;
46 :   end
47 :
48 : //COMPARATOR FOR ATTENDANCE SAFETY AND FA CALCULATOR
49 : always @(posedge clk) begin
50 :   if((total_attendance + to_attend) >= 26 )
51 :     fa = 1;
52 :   FAfa;
53 :   count2 = current_attendance << 2;
54 :   count1 = (total_attendance << 1) + total_attendance;
55 :   if(count2 >= count1)
56 :     is_safe = 1;
57 :   else
58 :     is_safe = 0;
59 : end
60 :
61 : //TOTAL NUMBER OF LEAVES CALCULATOR
62 : always @(posedge clk)begin
63 :   if(is_safe)begin
64 :     to_attend = 0;
65 :     if((current_attendance<<2) >= ((total_attendance<<1) + total_attendance))begin
66 :       temp_leaves = (current_attendance<<2) - ((total_attendance<<1) + total_attendance);
67 :       leaves = temp_leaves/3;
68 :     end
69 :   else
70 :     leaves = 0;
71 : end
72 :
73 : //TOTAL NUMBER OF DAYS TO ATTEND CLASSES TO BE SAFE CALCULATOR
74 : always @(*)
75 :   begin
76 :     leaves = 0;
77 :     if(((total_attendance<<1) + total_attendance) >= (current_attendance<<2))
78 :       to_attend = ((total_attendance<<1) + total_attendance) - (current_attendance<<2);
79 :     else
80 :       to_attend=0;
81 :   end
82 :
83 : endmodule

```

Tcl Console Messages Log

DSD\_lab\_project - [C:/Users/User/DS... - Vivado 2023.1

File Edit Flow Tools Reports Window Layout View Run Help Q: Quick Access

Ready Default Layout

**Flow Navigator**

- PROJECT MANAGER
  - Settings
  - Add Sources
  - Language Templates
  - IP Catalog
- IP INTEGRATOR
  - Create Block Design
  - Open Block Design
  - Generate Block Design
- SIMULATION**
  - Run Simulation
- RTL ANALYSIS
  - Run Linter
  - Open Elaborated Design
- SYNTHESIS
  - Run Synthesis
  - Open Synthesized Design
- IMPLEMENTATION
  - Run Implementation
  - Open Implemented Design
- PROGRAM AND DEBUG
  - Generate Bitstream
  - Open Hardware Manager

**SIMULATION - Behavioral Simulation - Functional - sim\_1 - tb\_attendance\_register**

attendance\_register.v \* tb\_attendance\_register.v \* Untitled 39\* x

C:/Users/User/DS.../srcs\_1/new/attendance\_register.v

```

50 :   if((total_attendance + to_attend) >= 26 )
51 :     fa = 1;
52 :   FAfa;
53 :   count2 = current_attendance << 2;
54 :   count1 = (total_attendance << 1) + total_attendance;
55 :   if(count2 >= count1)
56 :     is_safe = 1;
57 :   else
58 :     is_safe = 0;
59 : end
60 :
61 : //TOTAL NUMBER OF LEAVES CALCULATOR
62 : always @(posedge clk)begin
63 :   if(is_safe)begin
64 :     to_attend = 0;
65 :     if((current_attendance<<2) >= ((total_attendance<<1) + total_attendance))begin
66 :       temp_leaves = (current_attendance<<2) - ((total_attendance<<1) + total_attendance);
67 :       leaves = temp_leaves/3;
68 :     end
69 :   else
70 :     leaves = 0;
71 : end
72 :
73 : //TOTAL NUMBER OF DAYS TO ATTEND CLASSES TO BE SAFE CALCULATOR
74 : always @(*)
75 :   begin
76 :     leaves = 0;
77 :     if(((total_attendance<<1) + total_attendance) >= (current_attendance<<2))
78 :       to_attend = ((total_attendance<<1) + total_attendance) - (current_attendance<<2);
79 :     else
80 :       to_attend=0;
81 :   end
82 :
83 : endmodule

```

Tcl Console Messages Log

## THE TESTBENCH:

DSD\_lab\_project - [C:/Users/User/DSD\_lab\_project/DSD\_lab\_project.xpr] - Vivado 2023.1

File Edit Flow Tools Reports Window Layout View Run Help Q: Quick Access 10 us Default Layout Ready

**Flow Navigator** SIMULATION - Behavioral Simulation - Functional - sim\_1 - tb\_attendance\_register

**PROJECT MANAGER**

- Settings
- Add Sources
- Language Templates
- IP Catalog

**IP INTEGRATOR**

- Create Block Design
- Open Block Design
- Generate Block Design

**SIMULATION**

- Run Simulation

**RTL ANALYSIS**

- Run Linter
- Open Elaborated Design

**SYNTHESIS**

- Run Synthesis
- Open Synthesized Design

**IMPLEMENTATION**

- Run Implementation
- Open Implemented Design

**PROGRAM AND DEBUG**

- Generate Bitstream
- Open Hardware Manager

attendance\_register.v tb\_attendance\_register.v Untitled 39\*

C:/Users/User/DSD\_lab\_project/DSD\_lab\_project.srscs/sim\_1/new/tb\_attendance\_register.v

```

1 //timescale 1ns / 1ps
2 /////////////////////////////////////////////////
3 // TEAMATES:NISHCHAY PALLAV           YASH
4 // ROLL NUMBERS:221ECC233            221ECC263
5 //
6 // Create Date: 13.11.2023 18:27:00
7 // Module Name: attendance_register
8 // Project Name: ATTENDANCE REGISTER
9 // Tool Versions: 1.0
10 // Description: A USER FRIENDLY ATTENDANCE REGISTER FOR STUDENTS
11 //
12 //
13 // Revision 0.01 - File Created
14 // Additional Comments: LAB PROJECT FOR 3 SEMESTER DGD LAB EC204
15 /////////////////////////////////////////////////
16
17 module tb_attendance_register;
18     // Inputs
19     reg clk;
20     reg attendance;
21     reg clr;
22
23     // Outputs
24     wire is_safe;
25     wire FA;
26     wire [3:0]leaves;
27     wire [6:0]to_attend;
28     wire [6:0]total_attendance;
29     wire [6:0]current_attendance;
30
31     // Attendance register instance
32     attendance_register uut_attendance_checker(
33         .clk(clk),
34         .clr(clr),
35         .attendance(attendance),
36         .is_safe(is_safe),
37     );
38
39     initial begin
40         clk <= 1'b0;
41         attendance=1;clr <= 1'b1;#1
42         attendance<=1;clr <= 1'b0;#11
43         attendance<=0;clr <= 1'b0;#7
44         attendance<=1;clr <= 1'b0;#6
45         attendance=0;
46     end
47
48     // Clock generator
49     always #0.5 clk <= ~clk;
50
51     endmodule
52
53
54
55
56
57
58

```

Tcl Console Messages Log 24:13 Insert Verilog

DSD\_lab\_project - [C:/Users/User/DSD\_lab\_project/DSD\_lab\_project.xpr] - Vivado 2023.1

File Edit Flow Tools Reports Window Layout View Run Help Q: Quick Access 10 us Default Layout Ready

**Flow Navigator** SIMULATION - Behavioral Simulation - Functional - sim\_1 - tb\_attendance\_register

**PROJECT MANAGER**

- Settings
- Add Sources
- Language Templates
- IP Catalog

**IP INTEGRATOR**

- Create Block Design
- Open Block Design
- Generate Block Design

**SIMULATION**

- Run Simulation

**RTL ANALYSIS**

- Run Linter
- Open Elaborated Design

**SYNTHESIS**

- Run Synthesis
- Open Synthesized Design

**IMPLEMENTATION**

- Run Implementation
- Open Implemented Design

**PROGRAM AND DEBUG**

- Generate Bitstream
- Open Hardware Manager

attendance\_register.v tb\_attendance\_register.v Untitled 39\*

C:/Users/User/DSD\_lab\_project/DSD\_lab\_project.srscs/sim\_1/new/tb\_attendance\_register.v

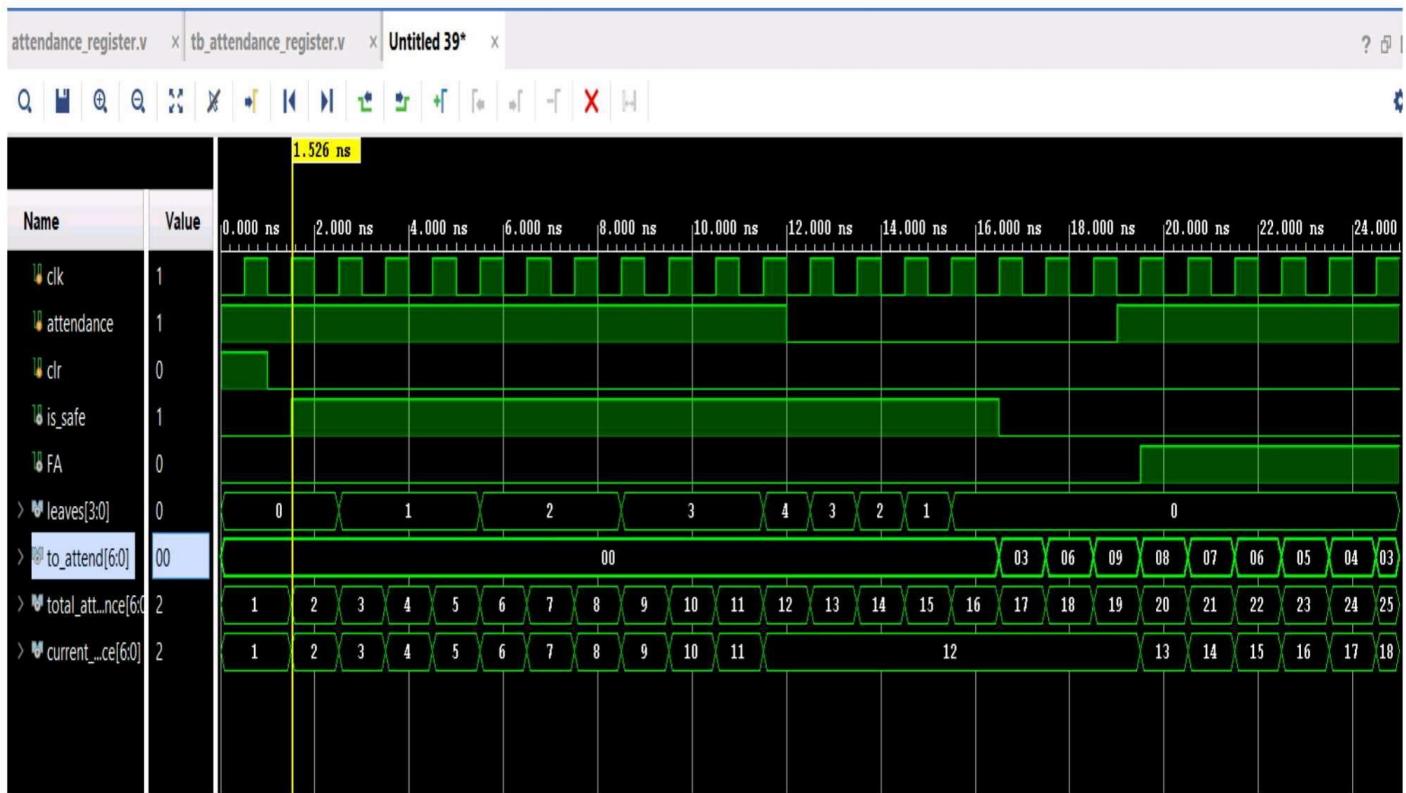
```

22 reg clk;
23
24 // Outputs
25 wire is_safe;
26 wire FA;
27 wire [3:0]leaves;
28 wire [6:0]to_attend;
29 wire [6:0]total_attendance;
30 wire [6:0]current_attendance;
31
32 // Attendance register instance
33 attendance_register uut_attendance_checker(
34     .clk(clk),
35     .clr(clr),
36     .attendance(attendance),
37     .is_safe(is_safe),
38     .FA(FA),
39     .leaves(leaves),
40     .to_attend(to_attend),
41     .total_attendance(total_attendance),
42     .current_attendance(current_attendance)
43 );
44
45 initial begin
46     clk <= 1'b0;
47     attendance=1;clr <= 1'b1;#1
48     attendance<=1;clr <= 1'b0;#11
49     attendance<=0;clr <= 1'b0;#7
50     attendance<=1;clr <= 1'b0;#6
51     attendance=0;
52 end
53
54 // Clock generator
55 always #0.5 clk <= ~clk;
56
57 endmodule
58

```

Tcl Console Messages Log 24:13 Insert Verilog

## SIMULATION:

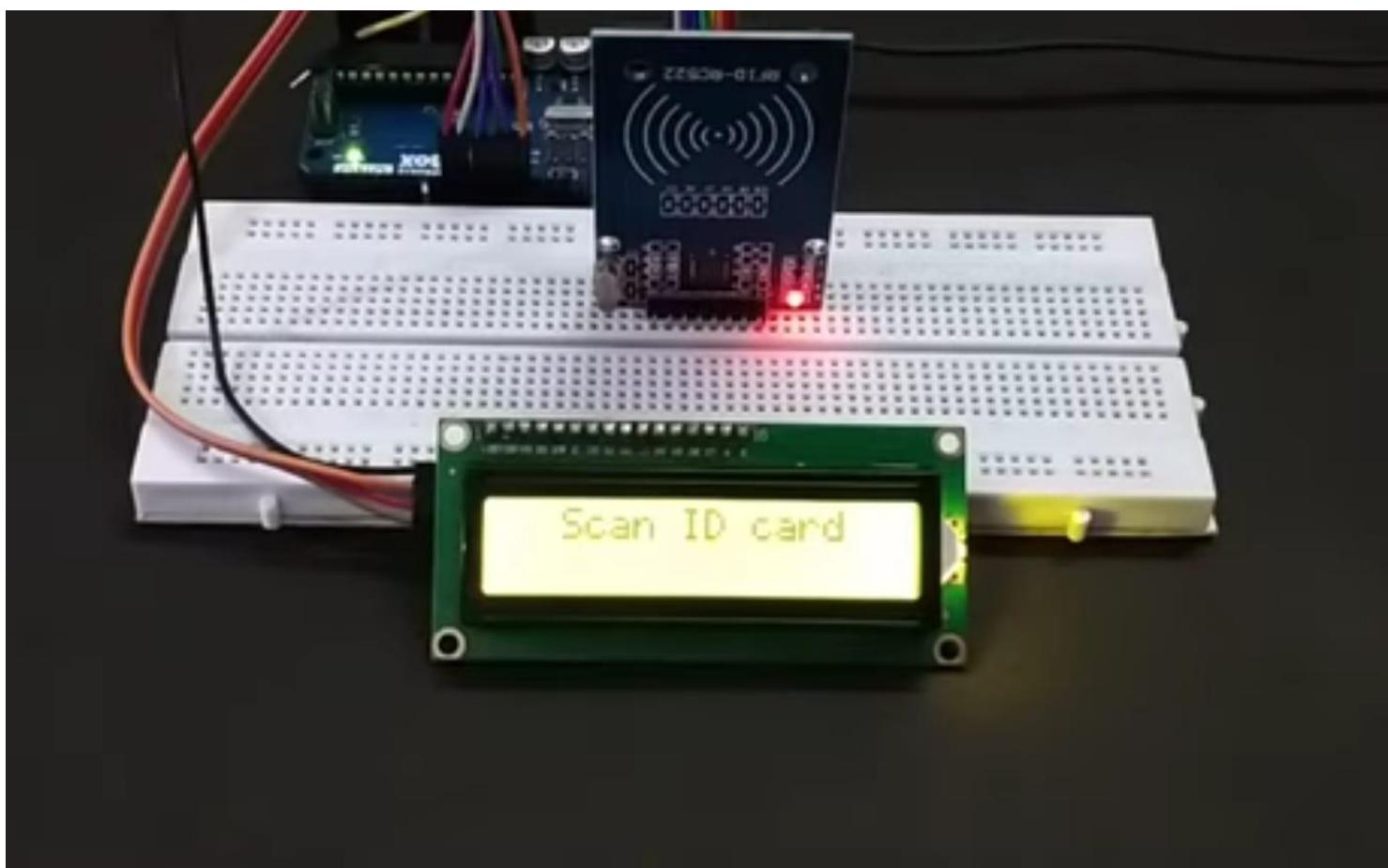


## **HARWARE OUTPUT:**

In this phase we have designed a hardware component by converting the Verilog code into Arduino code and stored it in the disc.

### **Here we have used following component:**

1. Arduino board
2. RFID card
3. Breadboard
4. LCD display
- 5. Jumper wires**



## WORKING:

1. The code uses the clock pulses to describe a passage of a day, this clock is used as input to the total attendance counter: whenever a day passes i.e., with each cycle of the clock pulse the counter increase its value by 1, the value of the total attendance is capped at 25 days.
2. After which the code is reset using a defined reset statement that signifies the start of new batch, this is done using if statement: total attendance $\geq$ 26, the batch can also be reset using an input called clear.
3. The current attendance counter works by increasing its count if attendance input is high i.e., the student is present and a day passes i.e., a clock pulse.
4. The percentage criterion of 75% is calculated and compared using the idea of multiplying the total attendance by 3 and current attendance by 4 derived from the statement:  $(\text{total attendance}/\text{current attendance}) \geq 3/4$  (75%). This is done by shifting the bit value of current attendance twice in left direction:  $(\text{total attendance} \ll 2)$ , and bit value of total attendance once in left direction and adding itself to the shifted value:  $(\text{current attendance} \ll 1 + \text{current attendance})$ .
5. Using the same method for multiplying by 3 and 4 we can calculate the values for leaves that is found from the equation:  $(\text{current attendance}/ (\text{total attendance} + \text{leaves})) = 3/4$ . This solves to a final equation:  
$$\text{leaves} = (4 * \text{current attendance} - 3 * \text{total attendance})/3.$$
6. Similarly, we find the value of to attend using the equation:  
$$(\text{current attendance} + \text{to attend})/ (\text{total attendance} + \text{to attend}) = 3/4$$
. This solves to a final equation:  
$$\text{To attend} = 3 * \text{total\_attendance} - 4 * \text{current attendance}.$$
7. To save us from the cases where the value of leaves and to attend are negative we use if statements with suitable statements.
8. Finally the FA criterion is decided when the remaining number of days the classes are left are less than the numbers of days you have to attend by the relation:  
$$(\text{total attendance} + \text{to attend}) \geq 26.$$

## **SUMMARY:**

### **Project Objective**

- Develop a digital attendance checker for students.
- Utilizes Verilog design for software or circuit implementation.
- Records individual attendance and compares it with a predetermined threshold.
- Aims to provide timely intervention and support in educational settings.

### **Components Used (If hardware implemented):**

- Flip-flop based Counters.
- Shift Registers.
- 4 Bit Adders.
- 4 Bit Comparators.
- 7-Segment Display.
- Basic gates

### **Functionality:**

- Records attendance using up-down counters and flip-flops.
- Utilizes Shift Registers, 4-Bit Adders, and 4-Bit Comparators for mathematical operations.
- Designed for a maximum of 25 sessions.
- Students labelled safe if attendance is above 75%, otherwise unsafe.

### **Problem Statement:**

- Device includes current attendance, total days, leave information, and safety status.
- Addresses attendance concerns in educational institutions and functional bodies.
- Enables timely actions and maintains manageable attendance records.

### **Counting Attendance:**

- Uses up counters with sequentially arranged flip-flops.
- Involves comparison with a threshold using Shift Registers, 4-Bit Adders, and 4-Bit Comparators.

### **Design:**

- Flowchart guides the working of the device.
- Inputs include clock (clock), clear (clear), and attendance.
- Outputs include total\_attendance, current\_attendance, is\_safe, FA, leaves, and to\_attend.

## **Working:**

1. Clock pulses represent a day, increasing total\_attendance.
2. Reset occurs after 25 days or through a clear input.
3. current\_attendance increases on student attendance.
4. Safety status is determined by comparing attendance percentages.
5. Leaves and to\_attend calculated to maintain safety status.
6. FA status determined based on remaining days and attendance requirements.

## **Equations Used:**

- $(\text{current\_attendance} / (\text{total\_attendance} + \text{leaves})) = 3/4$
- $(\text{current\_attendance} + \text{to\_attend}) / (\text{total\_attendance} + \text{to\_attend}) = 3/4$

## **Final Criteria:**

- FA status is determined when remaining days are insufficient to meet attendance requirements.

## **CONCLUSION:**

In conclusion, the developed digital attendance checker offers a robust solution for tracking and managing student attendance. By integrating various components such as counters, shift registers, and 4-bit comparators, the system provides a reliable mechanism to assess individual attendance against a predetermined threshold. The project not only addresses the immediate need for accurate attendance records in educational institutions but also highlights its potential applicability in diverse functional bodies. The inclusion of features like safety status, leave calculations, and future attendance requirements enhances its utility. Overall, this project serves as an effective tool for promoting timely intervention, facilitating proper program evaluation, and ensuring the seamless functioning of educational and corporate environments.

## **REFERENCES:**

- Verilog reference Appendix A.
- Digital system Design Lab record book.
- Google for basic understanding.
- IRIS attendance interface for idea.

[https://www.researchgate.net/publication/362405196\\_A\\_Literature\\_Review\\_on\\_Smart\\_Attendance\\_Systems](https://www.researchgate.net/publication/362405196_A_Literature_Review_on_Smart_Attendance_Systems)

<https://images.app.goo.gl/s9uDR3ufbBWw6pzq7>