# PROJECT: CLOCK WITH ALARM

**Group members:**
S.Sameera Tasneem (BU21EECE0100100)
N.Bhargavi (BU21EECE0100501)
V.Varshini (BU22EECE0100206)
S.Akhil (BU21EECE0100411)
K.Vamsi (BU21EECE0100085)

**Under the guidance of:**

**DR. ARUN KUMAR MANOHARAN**

# Introduction:

This mini project focuses on designing and implementing a digital clock with an alarm feature using a Field-Programmable Gate Array (FPGA). FPGAs are versatile, reconfigurable devices that offer high performance and parallel processing capabilities, making them ideal for various digital system applications. The project aims to leverage these features to create a functional and reliable clock with an alarm system.

The primary objectives are to design a digital clock that accurately displays hours, minutes, and seconds, and to integrate an alarm system that triggers an alert at a specified time. Additionally, a user-friendly interface will be developed to facilitate setting the time and alarm, as well as turning the alarm on and off. This project not only provides practical experience in FPGA programming but also covers essential aspects of digital electronics, including counters, state machines, and user interfaces.

Key tools and components for this project include an FPGA development board (such as Xilinx Vivado), Verilog hardware description language and user interface components like push buttons and 7-segment displays. The project scope encompasses designing timekeeping and alarm modules, handling user inputs, creating display modules, and integrating and testing the complete system on the FPGA.

Overall, this project offers hands-on experience in digital system design and FPGA implementation, culminating in a fully functional digital clock with an alarm feature. It serves as a valuable introduction to FPGA-based development and digital electronics.

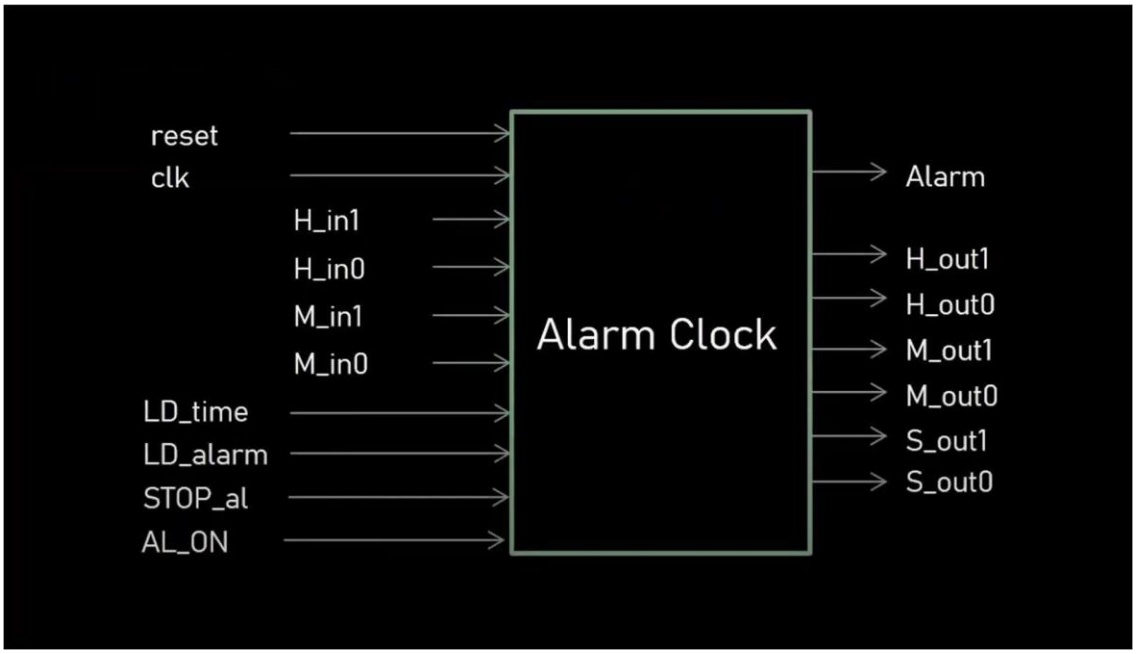
# Objectives :

- To Design a Digital Clock: Implement a digital clock that accurately displays hours, minutes, and seconds.
- To Develop an alarm system that can be set to a specific time and trigger an alert when the set time is reached.
- To Create a user-friendly interface for setting the clock time and alarm time, and for turning the alarm on and off.
- To Efficiently use FPGA resources to manage timekeeping and alarm functionalities.
- To Ensure the design is modular to facilitate easier testing, debugging, and potential future enhancements.
- Use simulation tools to verify the correctness of the design before hardware implementation.
- To Synthesize the design using FPGA development tools and implement it on an FPGA development board.
- To Test the integrated system on the FPGA board to ensure accuracy, reliability, and user interface responsiveness.

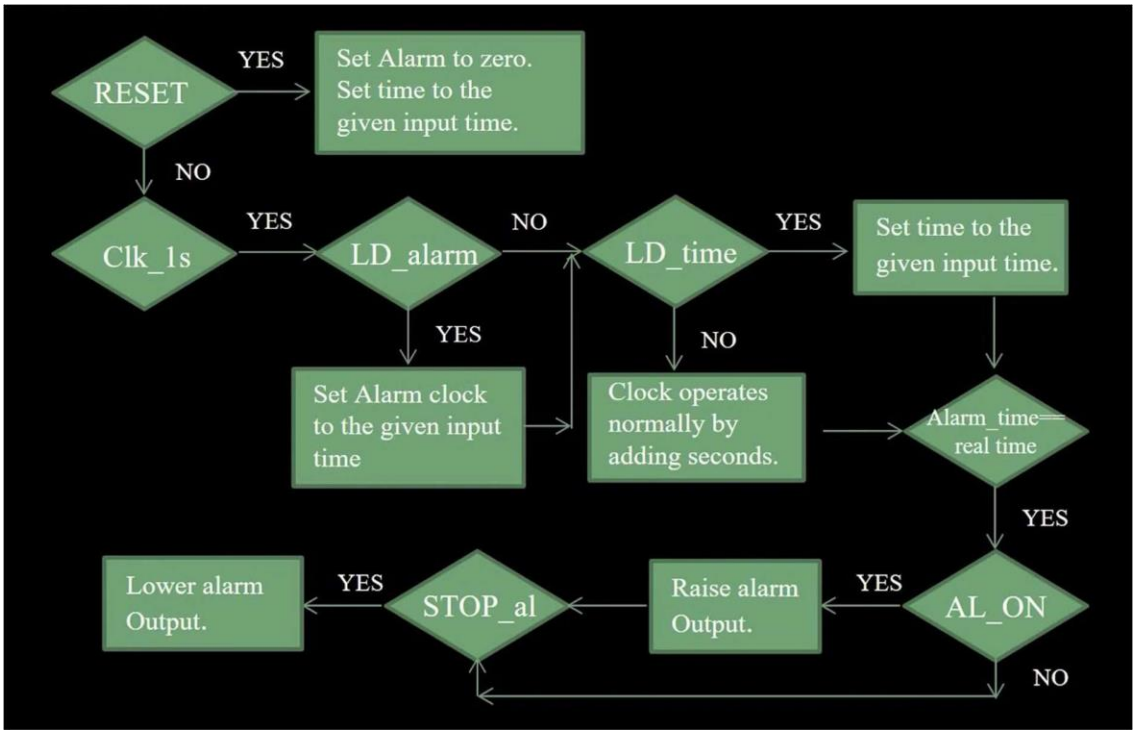
# Methodology/Apparatus required :

- Software Used:
  xilinx Vivado 2016.2
- Hardware used:
  Nexys 4: FPGA Trainer Board

- Block Diagram :



# Flowchart:




Steps to be followed for implementation Clock With alarm using FPGA kit:

- Open Xilinx Vivado software>create new project>create File>give file name (module name)
- Select Product category: General Purpose
- Family: Artix-7
- Package:cg324
- Speed value: -1
- Select Xc7a100tcsg324-1

## Enter the Code:

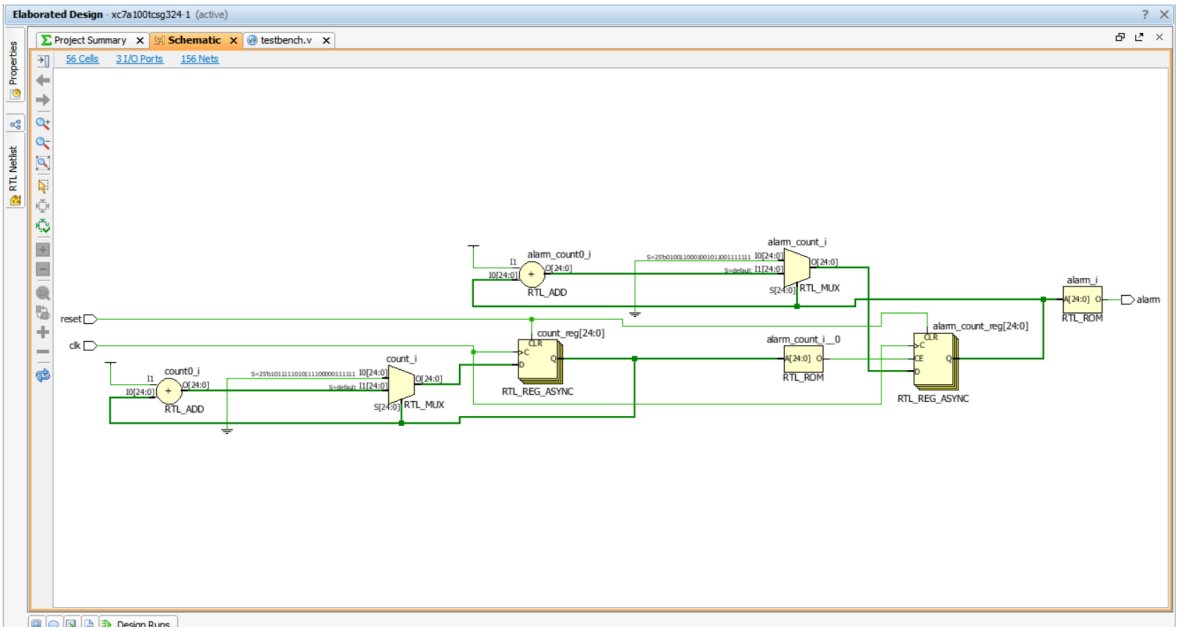
```
module   clock_with_alarm(
    input clk,         // Clock input
    input reset,       // Reset input
    output reg alarm    // Alarm output
);

reg [24:0] count;      // 25-bit counter to divide the clock frequency
reg [24:0] alarm_count; // Counter for the alarm

// Counter to divide the clock frequency to get 1 Hz clock
always @(posedge clk or posedge reset)
begin
    if (reset)
        count <= 0;
    else if (count == 25000000 - 1) // For 50 MHz input clock
        count <= 0;
    else
        count <= count + 1;
end
// Counter for the alarm
always @(posedge clk or posedge reset)
begin
    if (reset)
        alarm_count <= 0;
    else if (count == 25000000 - 1) // For 50 MHz input clock
    begin
        if (alarm_count == 10000000 - 1) // For 10 seconds
            alarm_count <= 0;
        else
            alarm_count <= alarm_count + 1;
    end
end
// Alarm logic
always @(*)
begin
    if (alarm_count == 10000000 - 1) // For 10 seconds
        alarm <= 1;
    else
        alarm <= 0;
end
endmodule
```

## Test Bench code:

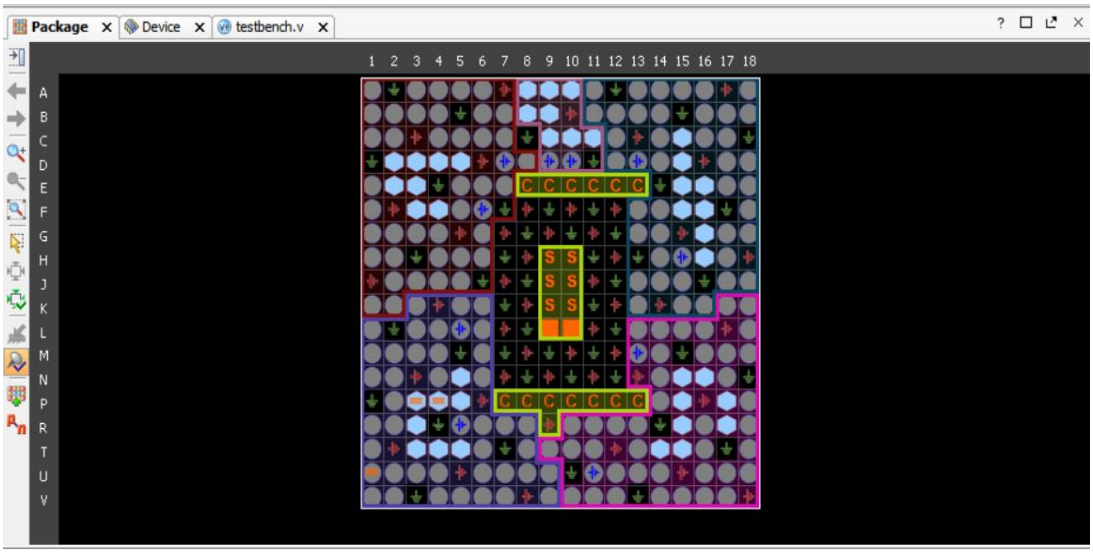
```
module clock_with_alarm_tb;

    // Inputs
    reg clk;
    reg reset;

    // Outputs
    wire alarm;

    // Instantiate the clock_with_alarm module
    clock_with_alarm dut (
        .clk(clk),
        .reset(reset),
        .alarm(alarm)
    );

    // Clock generation
    always begin
        clk = 0;
        #5;
        clk = 1;
        #5;
    end

    // Reset generation
    initial begin
        reset = 1;
        #10;
        reset = 0;
        #10000; // Simulate for a while
        $stop; // Stop simulation
    end

endmodule
```

- Go to Run simulation>Run behavioral simulation>Give force constant and force clock to inputs i.e., binary values

In RTL analysis>open elaborated design




- In schematic window, on the top change the “default Layout” to “I/O planning”
- Go to I/O ports[bottom panel]>select scalar ports
- Give I/O std as LMCMOS33

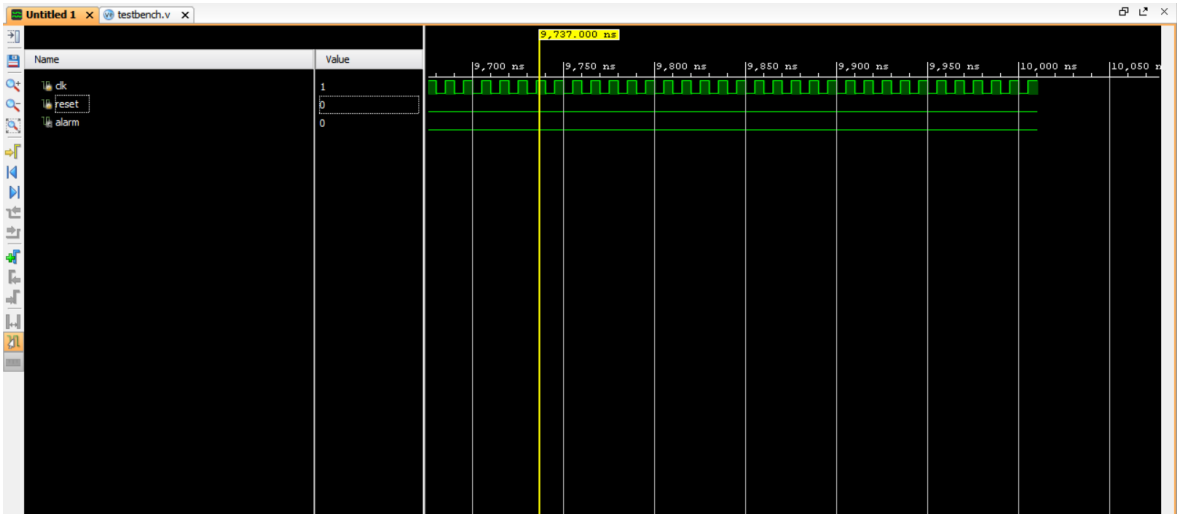



- Go to package pins>Give input and outputs

- Select save constraint[below edit button]
- From flow navigator>select Run suynthesis
- After successful synthesis>A pop appears>select Run implementation>click ok
- After finishing, another pop up appears, before selecting any option; connect the FPGA kit to CPU
- After Bit stream generation>choose open implemented design
- In flow navigator>program and debug>open hardware manager

  At top [green bar]>select open target>Auto connect
  At same bar>Program device>select the one which is shown>program

## Results



# VIDEO LINK:

https://drive.google.com/file/d/19V6R2GMuW8MmhADmGs4qveYNYwlUU1eb/view?usp=sharing