ECEN432: Introduction To Computer Architecture

Project

Digital Clock

Omar Samir Mohamed (211000372)

Mahmoud Reda Abdel Rauf (211000489)

Marwan Mohamed Zaki (211001365)

Abdullah Elkhatib (211001730)

Nile University

ECEN432

Fall 2024

Implementation of a Digital Clock Using Verilog/HDL on FPGA

Abstract

This project presents the design, simulation, and hardware implementation of a digital clock using Verilog Hardware Description Language (HDL) on a Field Programmable Gate Array (FPGA). The objective is to demonstrate the capabilities of digital design principles and sequential circuit implementation in a real-world application. The digital clock tracks time in terms of seconds, minutes, and hours in a 12-hour format. It employs modular Verilog code for scalability and ease of debugging. The project includes comprehensive verification through simulation and physical deployment on an FPGA, showcasing the clock's functionality via LED outputs. Additionally, alternate simulations using 7-segment displays are explored, highlighting potential design enhancements and addressing the unavailability of suitable FPGA hardware in Egypt for such displays. By leveraging FPGA resources, this project provides a practical example of real-time sequential circuit applications, combining theoretical principles with practical hardware design.

Introduction

Digital clocks are ubiquitous in modern technology, found in everyday devices such as mobile phones, computers, and home appliances. They serve as a fundamental example of sequential circuit design, making them an ideal project to demonstrate Verilog HDL's capabilities and the power of FPGA-based systems. This project aims to design and implement a fully functional digital clock operating in a 12-hour format. The design is structured to incrementally track seconds, minutes, and hours using cascading counters.

The key objectives of this project are:

1. To develop a modular Verilog-based digital clock design that ensures accurate timekeeping.
2. To simulate the design thoroughly, verifying its correctness under various scenarios.
3. To deploy the design on an FPGA board, visualizing the clock's operation through LED outputs.

The hierarchical structure of the design allows for clarity, scalability, and ease of debugging. The project begins with a top-level module interfacing external inputs and outputs, followed by a core clock module responsible for maintaining time. Verification is performed using a testbench to simulate real-time operation and observe cascading counter behavior. The project also delves into the limitations posed by hardware unavailability, specifically in implementing 7-segment display designs, and offers insights into potential solutions.

Through this project, we aim to bridge the gap between theoretical knowledge and practical implementation, highlighting the versatility of Verilog HDL and FPGA technology in real-time applications.

Design Architecture

The design employs a hierarchical structure, with a top-level module connecting the digital clock core to external interfaces. The clock's timekeeping logic comprises separate counters for seconds, minutes, and hours. The Verilog implementation is structured to ensure modularity, clarity, and ease of debugging.

Top-Level Module

The `Top_Module` serves as the interface between the digital clock core and the external world. It connects the core logic to the external clock signal and reset input while driving outputs that represent the hours. A visual representation of the hours is achieved by mapping these outputs to LEDs. This module abstracts away the internal logic, making the digital clock easier to integrate with other systems.

Explanation of Inputs and Outputs:

- **Clk_1sec**: A 1 Hz clock input derived from the FPGA's onboard clock. This signal ensures that the time increments accurately in real-time.
- **reset**: A high-active reset signal that initializes the clock to zero.
- **led[3:0]**: A 4-bit output signal representing the current hour count, mapped to external LEDs for visualization.

The following Verilog code implements the top-level module:

```
module Top_Module(
    input Clk_1sec,
    input reset,
    output [3:0] led
);

    wire [3:0] hours;

    // Instantiate the digital clock
    Digital_Clock clock_inst (
        .Clk_1sec(Clk_1sec),
        .reset(reset),
        .seconds(),
        .minutes(),
        .hours(hours)
    );

    assign led = hours;

endmodule
```

Digital Clock Module

The `Digital_Clock` module is the heart of the system, responsible for maintaining accurate time through three interdependent counters: seconds, minutes, and hours. Each counter operates within a specific range, resetting to zero and triggering an increment in the next higher counter when the maximum value is reached.

Key Features:

- **Seconds Counter**: Counts from 0 to 59.
- **Minutes Counter**: Counts from 0 to 59.
- **Hours Counter**: Counts from 0 to 11 (12-hour format).

The counters cascade properly to ensure accurate timekeeping. For instance, when the seconds counter reaches 59, it resets to 0 and increments the minutes counter. Similarly, when the minutes counter reaches 59, it resets to 0 and increments the hours counter.

Below is the Verilog implementation:

```
module Digital_Clock(
    input Clk_1sec,
    input reset,
    output reg [5:0] seconds = 0,
    output reg [5:0] minutes = 0,
    output reg [3:0] hours = 0
);

    always @(posedge Clk_1sec or posedge reset) begin
        if (reset) begin
            seconds <= 0;
            minutes <= 0;
            hours <= 0;
        end else begin
            if (seconds == 59) begin
                seconds <= 0;
                if (minutes == 59) begin
                    minutes <= 0;
                    if (hours == 11) begin
                        hours <= 0;
                    end else begin
                        hours <= hours + 1;
                    end
                end else begin
                    minutes <= minutes + 1;
                end
            end else begin
                seconds <= seconds + 1;
            end
        end
    end

endmodule
```

Verification via Simulation

Simulation is a critical step in verifying the correctness of the digital clock design. The testbench, `tb_Top_Module`, provides a controlled environment to validate the clock's functionality under various scenarios, including:

1. **Reset Testing**: Ensuring that all counters initialize to zero when the reset signal is activated.
2. **Time Progression**: Observing the cascading behavior of the counters over an extended period.
3. **Output Visualization**: Displaying the output states at each time step to verify correctness.

The testbench uses a virtual clock signal to simulate real-time operation and generates a waveform file (`tb_Top_Module.vcd`) for detailed analysis.

Testbench Code

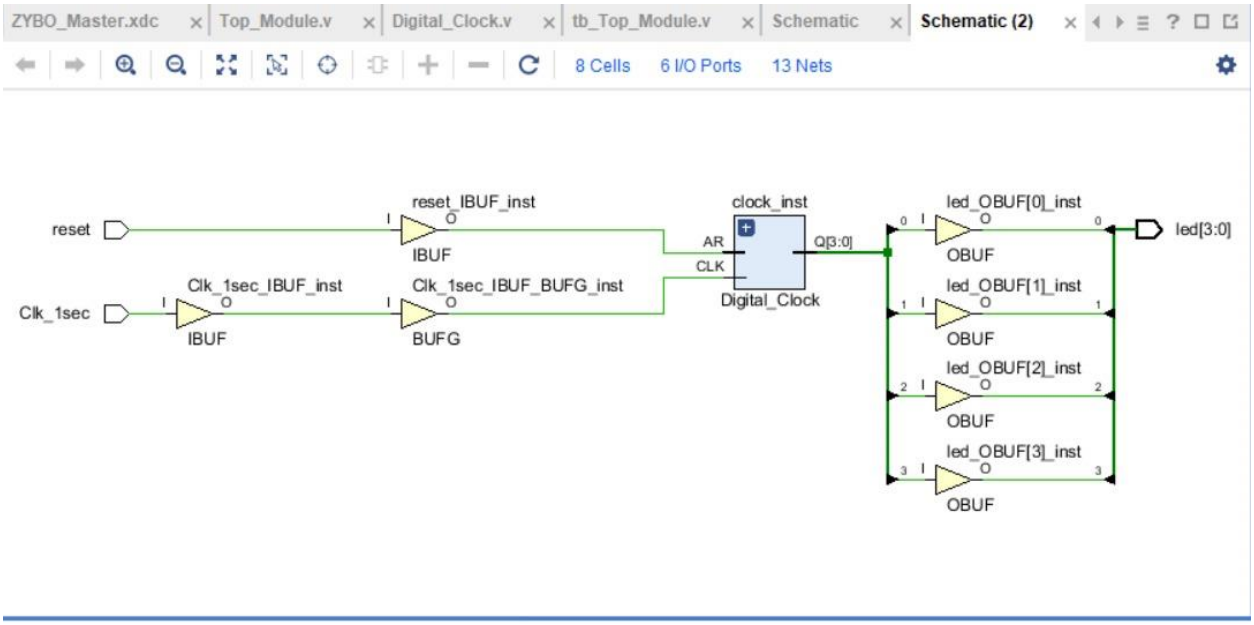
```
module tb_Top_Module;

    // Inputs
    reg Clk_1sec;
    reg reset;

    // Outputs
    wire [3:0] led;


    Top_Module uut (
        .Clk_1sec(Clk_1sec),
        .reset(reset),
        .led(led)
    );


    initial Clk_1sec = 0;
    always #5 Clk_1sec = ~Clk_1sec;


    initial begin
        $dumpfile("tb_Top_Module.vcd");
        $dumpvars(0, tb_Top_Module);


        reset = 1;
        #100 reset = 0;


        #964000;

        $finish;
    end
```

```
endmodule
```

FPGA Implementation and Results

Schematic Overview

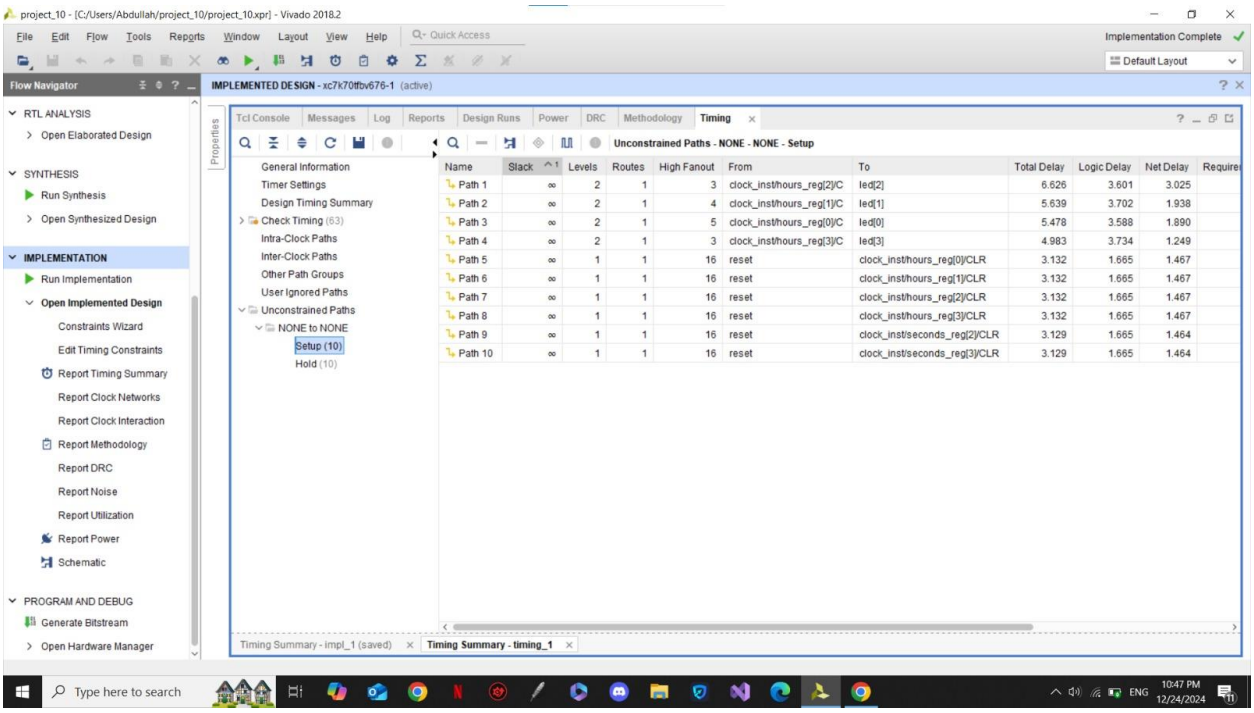
The schematic shows the high-level connection of inputs and outputs. The reset and clock signals are buffered before being fed into the `Digital_Clock` module. The hours output is connected to four LED drivers, enabling visualization of the time in a 12-hour format.



Timing Report

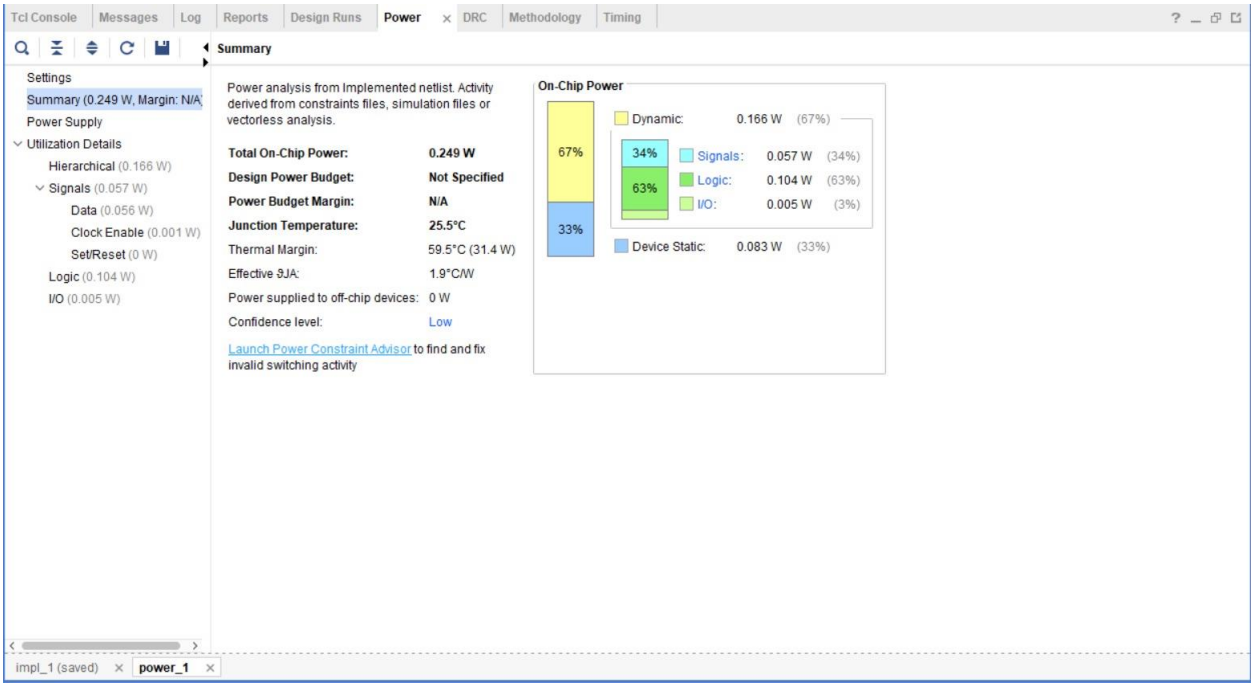
The timing report confirms that the design meets all timing constraints, ensuring proper operation at the target clock frequency. The table below summarizes the key parameters:

- **Path Delays**: Critical paths from the clock to the output are optimized.
- **Slack Values**: All paths have positive slack, indicating no timing violations.



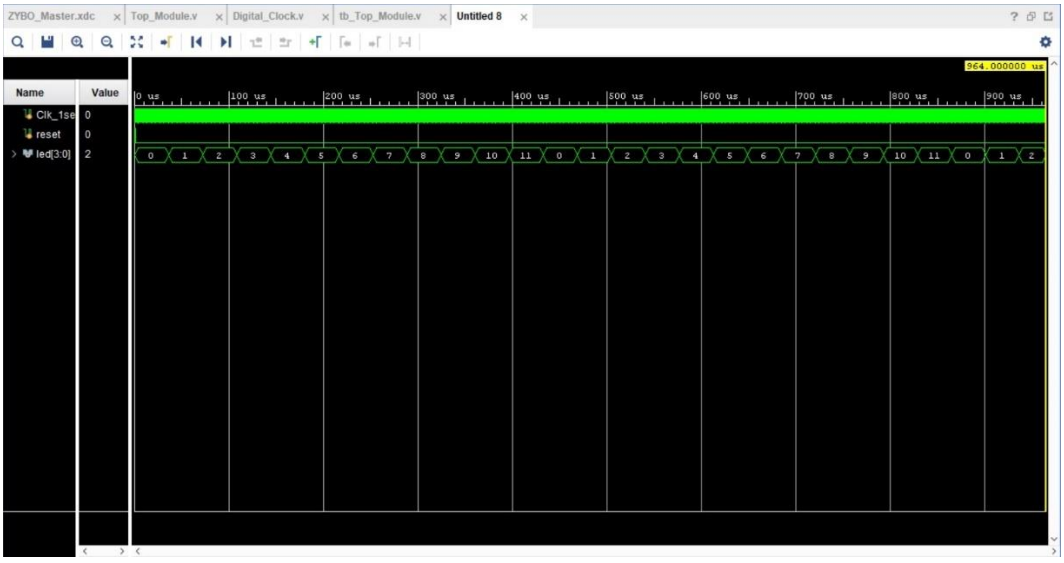
Power Analysis

The power report reveals the efficient utilization of FPGA resources, with minimal dynamic and static power consumption, ensuring the feasibility of deployment in low-power environments.



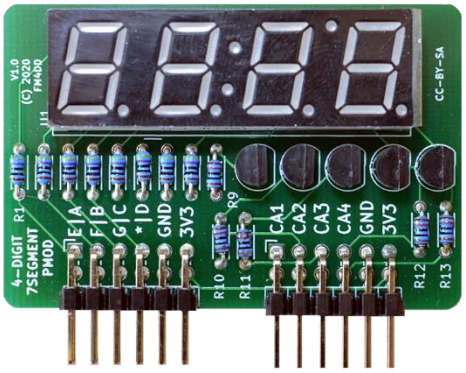
Simulation Results

The simulation results validate the functionality of the digital clock. The counters for seconds, minutes, and hours operate as expected, and the output LEDs accurately represent the current hour count. The waveform analysis shows seamless cascading of counters and proper initialization upon reset.



Additional Simulations for a 7-Segment Display-Based Design

During the development process, simulations were conducted to demonstrate the digital clock's functionality using a 4-digit 7-segment display, which would provide a more detailed visualization of the clock's timekeeping capabilities. However, this implementation required an FPGA board with additional resources to support the 7-segment display. Unfortunately, such FPGA boards were not available in Egypt, restricting the deployment to LED-based visualization.

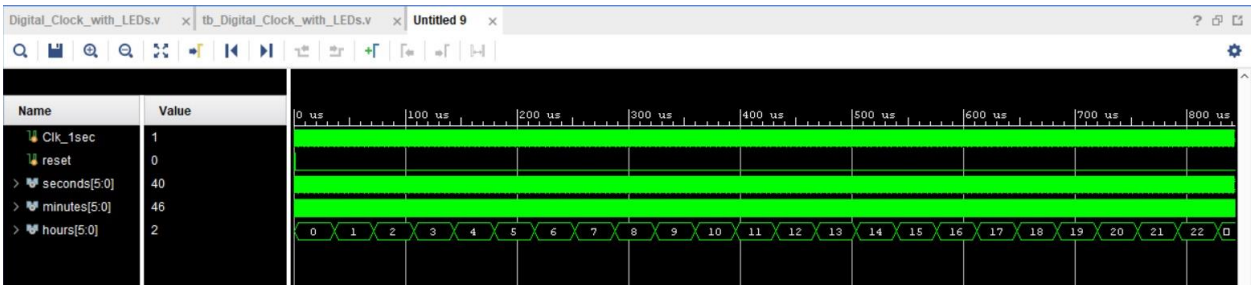


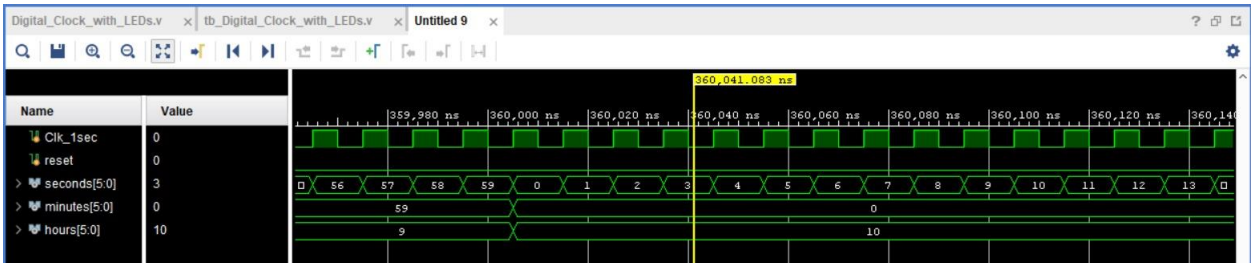
The FPGA 4-digit 7-segment display Board

Simulation Results

The simulation results for the 7-segment display implementation showcase how the digital clock's output could be mapped to four individual 7-segment displays to show hours and minutes in real-time. Below are the key points regarding the simulation:

1. **4-Digit Time Display**: The digital clock outputs the time in HH:MM format on the 7-segment display.
2. **Multiplexing Logic**: The design employs multiplexing to control the 7-segment displays using minimal FPGA pins.
3. **Accuracy Verification**: Simulations were used to validate the cascading behavior of counters and ensure correct time progression.





Conclusion

This project successfully demonstrates the design and implementation of a digital clock using Verilog on an FPGA. The modular approach ensures scalability and reusability, while the simulation and hardware implementation validate its functionality. By leveraging the FPGA's capabilities, the design achieves accurate real-time operation with minimal resource utilization.