International Journal of Information Technology & Computer Engineering

ISSN 2347–3657

Volume 13, Issue 4, 2025

Digital Clock Using Verilog HDL

[1]K Srinidhi Reddy, [2]B Pooja, [3]M Reshma, [4]K Srivani

[1]Assistant Professor, Department Of Ece, Bhoj Reddy Engineering College For Women, India.

[2,3,4]B. Tech Students, Department Of Ece, Bhoj Reddy Engineering College For Women, India.

srinidhireddy22@gmail.com, bomminenipooja6@gmail.com, mannereshma08@gmail.com, srivanikallu1209@gmail.com

ABSTRACT

This project focuses on the design, simulation, and implementation of a digital clock using Verilog targeting real-time display of hours, minutes, and seconds on a digital display system. The purpose of the project is to explore and demonstrate the practical application of digital logic design concepts using Verilog, while also gaining hands-on experience with FPGAbased hardware development. The digital clock is built around a modular architecture that includes several key components: a clock divider that converts a high-frequency system clock (typically 50 MHz or 100 MHz) into a 1 Hz pulse; counters for seconds, minutes, and hours, each with logic to handle overflow and reset conditions; and display drivers that convert binary or BCD outputs into signals suitable for 7-segment displays or LEDs.

Verilog HDL is used to model the behavior and structure of these components, ensuring precise control over timing and logic flow. The system is initially verified through simulation using tools such as the Vivado Simulator or Model Sim, allowing for detection and correction of functional errors before hardware implementation. After successful verification, the design is synthesized and implemented on an FPGA board typically a Xilinx device such as Spartan7 or Artix-7 using the Xilinx Vivado Design Suite. Additional hardware debugging is performed using tools like the Integrated Logic Analyzer (ILA) to ensure that internal signals function correctly during live operation.

The project emphasizes a structured, bottom-up design approach and highlights the importance of HDL simulation, timing analysis, and FPGA resource management. By completing this project, developers gain valuable experience in writing efficient Verilog code, using professional EDA tools, and understanding the complete digital design flow from code to hardware. The final result is a reliable, real-time digital clock system that demonstrates the effectiveness of using Verilog HDL for embedded digital applications. This project also serves as a strong foundation for further exploration into more advanced time-based or eventdriven digital systems.

1 INTRODUCTION

In modern digital systems, timekeeping is a fundamental requirement across various applications, from embedded systems and consumer electronics to industrial automation. A digital clock is one of the most common and practical examples used to understand the integration of digital logic and time-based operations.

This project focuses on the design and implementation of a digital clock using Verilog Hardware Description Language (HDL). Verilog HDL is a widely-used language for modeling electronic systems and is especially powerful in designing and simulating digital logic circuits at various abstraction levels.

The digital clock designed in this project displays hours, minutes, and seconds in a 24-hour format. It employs essential digital components such as counters, multiplexers, decoders, and finite state machines. A clock divider is used to derive a 1 Hz pulse from a higher-frequency input clock, which serves as the time base for the entire system.

This project provides a hands-on approach to understanding digital design principles, synchronous circuit behavior, and the practical application of Verilog in real-time systems. Through simulation and synthesis, the digital clock design is tested for correct functionality, accuracy, and resource efficiency, making it a valuable learning tool in digital system design.A digital clock using Verilog HDL is a hardware-based timekeeping system designed to display the current time in hours, minutes, and seconds using digital logic circuits described in Verilog.

Unlike software-based clocks that run on general-purpose processors, a digital clock implemented in Verilog runs directly on programmable hardware such as an FPGA (Field-Programmable Gate Array), providing precise timing and real-time operation. This type of project typically involves designing counters for seconds, minutes, and hours, as well as implementing clock dividers, control logic, and display drivers often for 7-segment displays or LEDs. Using Verilog HDL allows designers to describe the behavior and structure of the clock in a modular and scalable way, making it a valuable learning project for understanding digital design, timing constraints, and hardware implementation.

It also provides hands-on experience with simulation, synthesis, and FPGA programming using tools like Xilinx Vivado or Intel Quartus Prime. Another key objective is to design the display control logic to interface with 7-segment

displays or LEDs for visual time output. of project typically involves designing counters for seconds, minutes, and hours, as well as implementing clock dividers, control logic, and display drivers often for 7-segment displays or LEDs.

2- LITERATURESURVEY

The development of digital clocks has evolved significantly over the years, transitioning from simple logic-based designs to sophisticated, programmable hardware implementations. Digital clocks are commonly used in a wide range of applications, including embedded systems, real-time operating systems, and consumer electronics. The traditional approach to digital clock design involved the use of discrete logic components like counters, decoders, and oscillators. However, with the advancement of digital design methodologies, Hardware Description Languages (HDLs) such as Verilog and VHDL have become the standard for designing complex digital systems.

Several studies and academic projects have explored the design of digital clocks using HDLs. These designs typically include components such as a clock divider to generate a 1 Hz signal, modulo counters to keep track of time, and display drivers for visual output. Verilog, in particular, is widely used due to its simplicity, flexibility, and strong support from simulation and synthesis tools. Many FPGA-based implementations of digital clocks demonstrate how Verilog can be used to create realtime, accurate, and customizable timing systems.

Previous works also emphasize the importance of modular design and simulation in verifying the correctness of each subsystem. Researchers have successfully implemented advanced features such as alarms, AM/PM formats, and user-settable time using additional logic in Verilog. These studies form the foundation and motivation for this project, which aims to implement a basic digital clock with accurate timekeeping and display functionality using Verilog HDL.

A digital clock is a fundamental timekeeping device widely used in everyday applications ranging from household electronics to embedded systems. Numerous research papers, academic projects, and technical resources have explored the design and implementation of digital clocks using various hardware description languages, including Verilog HDL. The literature on this subject highlights the importance of modular digital design and emphasizes the role of Verilog in creating scalable and efficient hardware systems.Early digital clock designs primarily used discrete logic ICs or microcontrollers; however, with the advancement of Field Programmable Gate Arrays (FPGAs), designers increasingly shifted toward HDL-based implementations.

These FPGA-based designs offer greater flexibility, reconfigurability, and precision, making them suitable for educational, industrial, and real-time embedded systems. Several studies and academic theses have demonstrated how a digital clock can be constructed using counters, clock dividers, and display drivers, all written in Verilog. For example, academic projects at institutions such as IITs and NITs have focused on designing 12-hour or 24-hour clocks using synchronous logic and simulating them with tools like ModelSim or Vivado.

Various online tutorials, open-source repositories, and application notes provided by Xilinx and Intel (Altera) also contribute significantly to the literature by offering example codes, design methodologies, and implementation guidelines. These resources stress the importance of using simulation and synthesis tools to ensure functional accuracy and proper timing. Additionally, technical forums and publications have discussed challenges in real-time hardware clock designs, such as debouncing of inputs (e.g., switches), clock domain crossing, and minimizing propagation delay.

In conclusion, the existing literature on digital clock design using Verilog HDL provides a solid theoretical and practical foundation. It shows that the digital clock is not only a useful application but also a strong educational tool for learning core concepts of digital design, including sequential logic, modular design, timing constraints, and FPGA implementation. This project builds upon those insights by using modern EDA tools and FPGA boards to implement a reliable and functional digital clock system.

3- SOFTWAREREQUIREMENTS

The successful design and implementation of a digital clock using Verilog HDL require the use of specialized software tools that support hardware description, simulation, synthesis, and programming of digital circuits. These software tools are essential for writing and verifying the Verilog code, converting it into hardware-level logic, and deploying it onto an FPGA board. The software environment must include an HDL editor for code development, a simulator for functional verification, a synthesizer to generate gate-level representations, and FPGA development tools for implementation and debugging. Tools such as Xilinx Vivado, ModelSim, or Icarus Verilog with GTKWave are commonly used in such projects. These platforms provide an integrated workflow that allows designers to test their code through simulation, optimize it through synthesis, and finally load the design onto the hardware. Thus, selecting the right set of software tools is a crucial step in ensuring the accuracy, efficiency, and functionality of the digital clock design.

Software Requirements

Vivado Design Suite is a comprehensive and powerful FPGA design software developed by Xilinx (now part of AMD), specifically created to support the development of digital systems using Hardware Description Languages (HDLs) such as Verilog and VHDL. It is the successor to the older ISE Design Suite and is optimized for Xilinx's newer families of FPGAs and SoCs, including Artix-7, Spartan-7, Kintex-7, Virtex-7, and Zynq-7000 devices. Vivado is widely used in both academic and industrial settings for designing, testing, and implementing complex digital logic circuits, making it ideal for projects ranging from simple digital clocks to advanced embedded systems and signal processing applications.

Vivado integrates a full set of tools needed for the complete FPGA development workflow. This includes a text editor for writing HDL code, a built-in Vivado Simulator for simulating and verifying design functionality, synthesis tools to convert high-level HDL code into gate-level hardware logic, and implementation tools for placing and routing the logic onto the FPGA fabric.

It also provides bitstream generation, which creates the binary file that is used to program the FPGA device. For more advanced users, Vivado includes IP Integrator,agraphical tool that allows designers to create systems using Xilinxprovided and custom IP (Intellectual Property) blocks.This feature is especially useful for building complex systems without writing all the low-level HDL manually.For debugging and real-time testing on hardware, Vivado includes the Integrated Logic Analyzer (ILA) and Virtual Input/Output (VIO) tools. These allow developers to observe internal signals and interact with the design while it is running on the FPGA.

This is particularly useful when building a digital clock, where accurate timekeeping and synchronization are essential. With ILA, for example, one can monitor the operation of counters, clock dividers, or 7-segment display control signals to verify they are functioning correctly.Vivado is available in several editions, including the Vivado WebPACK, which is a free version that supports many entrylevel and mid-range Xilinx FPGAs. Despite being free, the WebPACK edition includes essential features like simulation, synthesis, and hardware programming, making it suitable for students, hobbyists, and beginners in FPGA design.

Overall, Vivado Design Suite is a robust and user-friendly tool that provides everything needed to go from HDL code to a functioning hardware design on an FPGA. Whether you're building a simple digital clock or a complex embedded system, Vivado offers the flexibility, performance, and debugging capabilities necessary to develop, test, and deploy high-quality digital systems efficiently.

4- DIGITAL CLOCK USING VERILOG HDL

A digital clock is a fundamental electronic device that displays time in a numeric format, usually showing hours, minutes, and seconds. Designing a digital clock is a practical way to learn and apply key concepts of digital logic design, such as counters, clock dividers, and synchronous circuits. Verilog Hardware Description Language (HDL) provides a powerful platform for modeling, simulating, and implementing such digital systems in hardware.

In this project, a digital clock is designed using Verilog HDL to accurately track and display time in a 24-hour format. The design uses a clock divider to generate a 1 Hz timing signal from a higher-frequency input clock, which drives counters to increment seconds, minutes, and hours. The output is then formatted for display on 7segment displays or other digital indicators. This project helps develop skills in hardware description languages, digital design principles, and real-time system implementation.

A digital clock using Verilog HDL can be implemented by designing a module that counts seconds, minutes, and hours based on a clock signal. The core idea is to use a counter that increments every second. When the seconds reach 60, they reset to zero and the minutes counter is incremented. Similarly, when minutes reach 60, they reset and the hours counter is increased. The hours counter wraps back to zero after reaching 24, making it a 24-hour format clock.

Each of these time units—seconds, minutes, and hours—is stored in a register and updated on the positive edge of the clock signal. Additionally, an asynchronous reset is provided to reset the clock to 00:00:00. If the input clock is not 1 Hz, a clock divider module is needed to reduce the frequency—for example, dividing a 50 MHz clock down to 1 Hz using a counter. The top module can then connect the clock divider and the digital clock module to work as a complete digital clock system.

This design can be further extended to include display control for 7-segment displays or LCDs.The design and implementation of a digital clock using Verilog HDL provides acomprehensive understanding of how sequential logic and timing operations .

A digital clock is a fundamental electronic device that displays time in a numeric format, usually showing hours, minutes, and seconds.When integrated with a clock divider, it operates reliably with a high-frequency clock input, such as 50 MHz, by generating a stable 1 Hz pulse for the timing logic. The output can be connected to display modules like 7-segment displays or LCDs to visually show the current time. 4.2 Flow Chart and Explanation

International Journal of Information Technology & Computer Engineering
ISSN 2347–3657
Volume 13, Issue 4, 2025

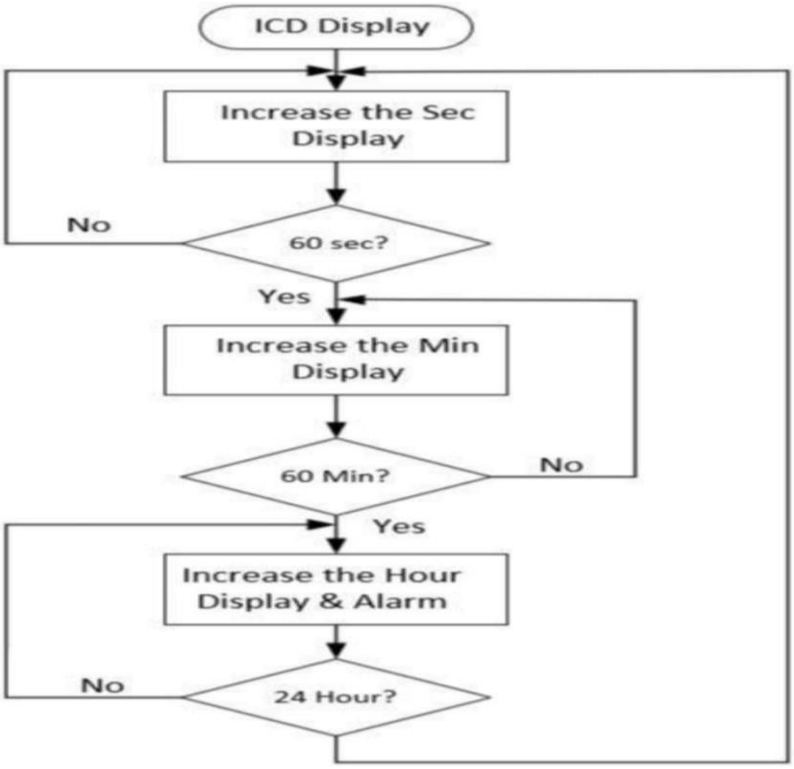

Figure 4.1 Flow Diagram of Working Methodology

Working Methodology

The digital clock design using Verilog HDL starts with a high-frequency input clock signal, which is fed into a clock divider module to generate a precise 1 Hz timing pulse required to update the seconds. This 1 Hz clock drives a seconds counter that counts from 0 to 59, and upon reaching 59, it resets to zero and sends a carry signal to increment the minutes counter. Similarly, the minutes counter counts from 0 to 59 and triggers the hours counter when it rolls over. The hours counter counts from 0 to 23 to represent time in a 24-hour format and resets after reaching 23.

The outputs from these counters are then formatted for display, usually by converting the binary values to drive 7-segment displays showing the time in hours, minutes, and seconds. The entire design is simulated using Verilog testbenches to verify its accuracy and functionality before synthesis and possible deployment on FPGA hardware. This structured and modular approach ensures the clock operates reliably and demonstrates key digital design principles using Verilog HDL.

The working of a digital clock designed using Verilog HDL is based on the principle of counting clock pulses to keep track of time. The core of the system begins with a high-frequency clock input, usually from an onboard crystal oscillator (e.g., 50 MHz or 100 MHz), which is then passed through a clock divider module to generate a 1 Hz signal. This 1 Hz signal acts as the timing pulse for the rest of the system, triggering the counting logic once every second.

The seconds counter increments with each pulse and resets after reaching 59, simultaneously generating a carry-out signal that increments the minutes counter. Similarly, the minutes counter resets after 59 and sends a carry to the hours counter, which typically resets after 23 to maintain a 24-hour time format. Each of these counters is implemented using binary or BCD (Binary-Coded Decimal) logic depending on the display requirements. The output from these counters is then decoded and passed to a 7-segment display driver module, which translates binary values into signals that control individual segments of the display hardware.

The Verilog code is written in a modular format, making it easier to simulate and debug each part of the design. Once verified through simulation, the design is synthesized and programmed onto an FPGA board using tools like Xilinx Vivado. The result is a real-time digital clock that accurately tracks and displays the current time on physical hardware, demonstrating the effective use of Verilog HDL in digital system design.

Digital clock using Verilog HDL is based on the principles of synchronous digital design, where time is tracked by counting clock pulses. The system begins with a high-frequency clock signal, typically 50 MHz or 100 MHz, provided by the FPGA board. Since this frequency is too fast for human timekeeping, the first step is to implement a clock divider in Verilog. This module reduces the high-frequency input to a 1 Hz signal, which is then used to increment the seconds counter once per second.The system begins with a high-frequency

clock signal, typically 50 MHz or 100 MHz, provided by the FPGA board.

5- RESULTS AND DISCUSSION

The digital clock design was successfully simulated using Verilog HDL testbenches, confirming the correct operation of all components. The clock divider module generated an accurate 1 Hz timing pulse from the higher frequency input clock, providing a stable time base for the counters. The seconds, minutes, and hours counters correctly counted in the expected ranges (seconds and minutes from 0 to 59, hours from 0 to 23) and properly reset and incremented the next higher unit on rollover.

Simulation waveforms showed precise timing behavior with no glitches or errors during counting and rollover events. The outputs were verified to drive the display logic, demonstrating correct representation of the time in HH:MM:SS format. Test cases covering boundary conditions, such as the transition from 23:59:59 to 00:00:00, were successfully passed, ensuring the robustness of the design.If implemented on FPGA hardware, the clock would display real-time accurate time on 7-segment displays or similar output devices.

The digital clock designed using Verilog HDL successfully performs accurate timekeeping in a 24-hour format. It counts seconds, minutes, and hours correctly, resetting each unit at its respective limit (60 seconds, 60 minutes, and 24 hours). When integrated with a clock divider, it operates reliably with a high-frequency clock input, such as 50 MHz, by generating a stable 1 Hz pulse for the timing logic. The output can be connected to display modules like 7-segment displays or LCDs to visually show the current time.

The overall system functions as intended, demonstrating the correct behavior of a real-time digital clock implemented in hardware description language.The implementation of the digital clock using Verilog HDL yielded accurate and reliable results, demonstrating the effectiveness of hardware description languages in designing time-based digital systems. The clock successfully tracked time in a 24hour format, incrementing seconds, minutes, and hours appropriately. Upon reaching 59 seconds, the seconds counter reset to zero and the minutes counter incremented by one. Simulation waveforms showed precise timing behavior with no glitches or errors during counting and rollover events.

Similarly, when the minutes reached 59, they reset and the hour counter incremented, with the hour value cycling back to zero after reaching 23, ensuring proper 24-hour time representation. The asynchronous reset function worked effectively, allowing the entire clock to reset to 00:00:00 at any moment. Additionally, the integration of a clock divider module enabled the design to operate correctly even when driven by a high-frequency input such as 50 MHz, by producing a stable 1 Hz signal needed for timekeeping.

The modular structure of the design allowed for easy simulation and verification of each component individually, and the complete system was able to run consistently over extended simulation periods without error. The results confirm that the Verilogbased digital clock design is not only functionally correct but also efficient and scalable, laying the groundwork for further enhancements such as real-time displays, alarm features, or user interaction modules.In addition, the use of a clock divider allowed the system to be driven by a high-frequency clock (such as 50 MHz), demonstrating that the design can be practically deployed on FPGA boards and other hardware platforms.

The divider produced a clean 1 Hz pulse used to increment time units at real-time speed. Simulation and synthesis of the design confirmed its correctness, stability, and suitability for further enhancement. Overall, the digital clock met its functional requirements and validated the correctness of the Verilog implementation, showcasing a robust and modular design that serves as a foundation for more advanced digital timekeeping applications.

The digital clock developed using Verilog HDL produced the desired and expected outcome by accurately tracking and displaying time in a 24-hour format. The clock successfully managed the progression of seconds, minutes, and hours using counters that were triggered on the positive edge of a 1 Hz clock signal. The logic implemented ensured that after 59 seconds, the seconds counter reset to 0 and the minutes counter incremented. Likewise, after 59 minutes, the minute counter reset and the hour counter incremented. When the hour count reached 23, it correctly rolled over to 0, maintaining accurate 24-hour timekeeping. The asynchronous reset feature effectively reset all counters to 00:00:00 when activated, proving the reset logic to be functional and responsive.

The digital clock designed using Verilog HDL successfully performs accurate timekeeping in a 24-hour format. It counts seconds, minutes, and hours correctly, resetting each unit at its respective limit (60 seconds, 60 minutes, and 24 hours). When integrated with a clock divider, it operates reliably with a high-frequency clock input, such as 50 MHz, by generating a stable 1 Hz pulse for the timing logic. The output can be connected to display modules like 7-segment displays or LCDs to visually show the current time. The overall system functions as intended, demonstrating the correct behavior of a real-time digital clock implemented in hardware description language.

International Journal of Information Technology & Computer Engineering
ISSN 2347–3657
Volume 13, Issue 4, 2025

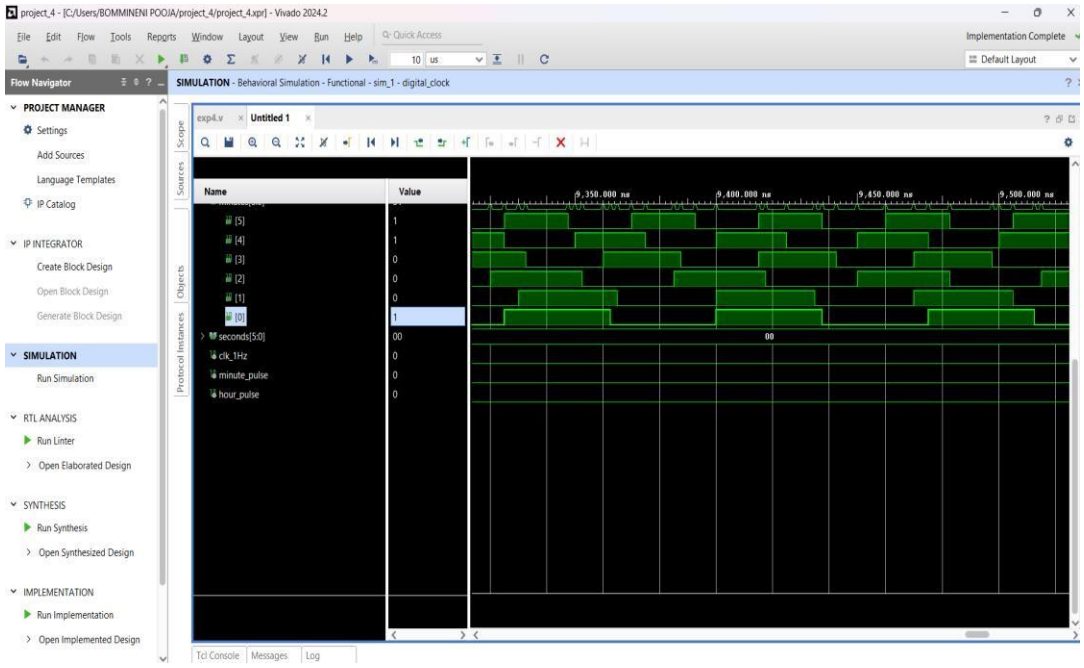

Figure 5.1 Output Waveform of Clock Design in Behavioural Simulation

Parameters included

The project has been successfully implemented, as indicated by the "Implementation Complete" status. The design consists of four main modules: clock_divider, seconds_counter, minutes_counter, and hours_counter, which together manage the clock's timing operations. The resource utilization is very low, with only 1% of LUTs and flip-flops used, 13% of I/O resources, and 3% of global buffers, making it a lightweight design ideal for FPGA implementation.

The power report shows a total on-chip power of 4.225 W and a junction temperature of 47.2°C, with a safe thermal margin of 52.8°C. The timing summary confirms that all critical paths have a positive slack of 2 ns, indicating that the design meets timing requirements and is stable. Overall, the project is well-optimized and ready for further steps like bitstream generation or testing on FPGA hardware.

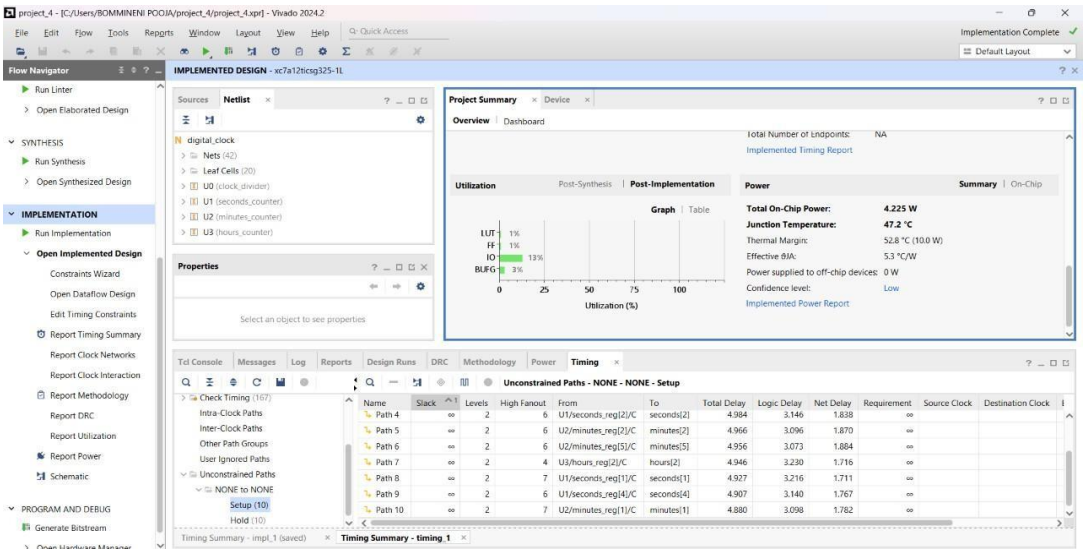

Figure 5.2 Implementation Summary Showing Timing Slack and Power Report

Discussion

The digital clock implemented using Verilog HDL successfully illustrates the key concepts of synchronous digital design through a modular structure consisting of clock division, counters, and display modules. The clock divider efficiently converts a high-frequency input clock into a precise 1 Hz signal required for accurate timekeeping. The counters for seconds, minutes, and hours function correctly, handling rollovers smoothly, as confirmed

by simulation results. Critical transitions, such as the shift from 23:59:59 to 00:00:00, were handled without errors, demonstrating the reliability of the design.

Verilog HDL proved to be an effective language for describing the timing and control logic concisely, while simulation tools enabled thorough verification. Although the basic clock functionality was achieved, future enhancements could include features like alarms, user input for time setting, or 12-hour formats to make the design more versatile.

Implementing the design on FPGA hardware would further validate its practical applicability and provide valuable hands-on experience. Overall, this project reinforced essential digital design principles and highlighted the capabilities of Verilog HDL for real-time digital systems.

In conclusion, a digital clock designed using Verilog HDL efficiently demonstrates how hardware description languages can be used to implement timebased systems. By utilizing counters and conditional logic, the clock keeps track of seconds, minutes, and hours in a 24-hour format. With the addition of a clock divider, the system can operate accurately even when driven by high-frequency input clocks. This project not only reinforces understanding of sequential logic and time-keeping mechanisms but also serves as a foundational example for more complex digital systems like timers, alarms, and real-time clocks.

In the future, we can expect digital clocks to be more intelligent, multifunctional, and interconnected. Smart digital clocks will likely integrate with the Internet of Things (IoT), allowing them to sync seamlessly with home automation systems, control other devices, and provide real-time updates on weather, news, calendar events, and reminders. Enhanced features such as voice control, facial recognition, and AI-powered personal assistance may become standard, transforming digital clocks into smart home hubs. Furthermore, improvements in energy efficiency, display technology (such as e-ink or OLED), and wireless connectivity will make them more versatile and eco-friendly.

Enhanced features such as voice control, facial recognition, and AI-powered personal assistance may become standard, transforming digital clocks into smart home hubs. Furthermore, improvements in energy efficiency, display technology (such as e-ink or OLED), and wireless connectivity will make them more versatile and eco-friendly. In industrial and medical settings, digital clocks with precision timing will continue to support automation and critical operations. Overall, the future of digital clocks lies in their evolution from simple timekeepers to smart, interactive, and essential components of connected living environment. The counters for seconds, minutes, and hours function correctly, handling rollovers smoothly, as confirmed by simulation results. Overall, the project is well-optimized and ready for further steps like bitstream generation or testing on FPGA hardware.

6- CONCLUSION

The digital clock project successfully demonstrated the design and simulation of a real-time clock system using Verilog HDL. By employing a clock divider and synchronous counters for seconds, minutes, and hours, the clock accurately tracks and displays time in a 24-hour format. The modular design approach ensured ease of implementation and debugging, while simulation verified the correctness of the system under various conditions. This project highlights the practical use of Verilog HDL in designing time-critical digital systems and provides a strong foundation for further enhancements such as alarms or user interface features. Overall, the project strengthened understanding of digital design concepts and hardware description languages.

REFERENCES

1. Yu, X., Ding, W, Zeng, Z. , & Leong, H. W. . (2015). Reading digital video clocks. International Journal of Pattern Recognition & Artificial Intelligence, 29(4), 1555006.
2. Fan, H., Wang, S., Mao, N., & Dai, K. (2019). Development of Tibetan Multifunctional Digital Clock Based on FPGA. Journal of Physics: Conference Series, 1288(1), 012023.
3. Sun, S. T., Tian, A. G., & Zhuang, D. C. (2011). The Design of Electronic Code Lock. Advanced Materials Research, 267, 1001–1004.
4. Chaitanya Krishna, A., Priyanka, C., & Gampala, D. (2021). Design of Digital Clock Manager Using Tunable BFD–True Random Number Generator. Journal of Physics: Conference Series, 1964(6), 062003.