Report: Dgital Alarm Clock Project

Marwan , Mazin Bersy, Yousef Sayed, Amr Eid

Department of Computer Science and Engineering, AUC

CSCE 230/2301 : Digital Design 1

Dr. Mohamed Shalan

May, 18 2023

Abstract

This report presents the development project of a digital alarm clock in digital design. Implemented on the BASYS3 board with the help of FPGA technology, this has helped us to enhance our problem-solving skills. Key features include clock division, counters, and 7 segment display , enabling the clock to display time and trigger alarms. The project involved designing distinct operational modes and overcoming challenges. Each team member contributed to different aspects of the project, emphasizing collaboration and hands-on learning. Overall, this project provided a valuable opportunity for us to apply theoretical knowledge to real-world problems in digital design.

Introduction

The digital clock creation is an essential educational project which aims to bring together the abstract concepts and practical aspects of digital design. FPGA (Field Programmable Gate Array) technology is a major crucial tool in modern electronics and digital systems. This FPGA project is a chosen essential coursework to build students' hands-on experience in this area. By engaging in this project, students not only reinforce their understanding of digital circuits, state machines, and timing issues but also develop essential skills in problem-solving, debugging, and system integration. This project aligns with the learning objectives of the Digital Design I course, providing an opportunity to apply classroom knowledge to a real-world problem.

In this project, the main objective was the designing, implementation and verification of a digital alarm clock on the basis of BASYS3 board. This meant making a working device through which both the hours as well as the minutes could be displayed onto a 7 segment LED and adding an alarm function that would sound when the time entered matched the predetermined alarm time. The task featured using the different peripherals like buttons and LEDs of the BASYS3 board to pull out user input and mode selection. We implemented two main operational modes: a normal clock mode for showing the current time and alarm time, as well as an adjust mode for setting the current time and alarm time of the watch. These characteristics were vital for securing well-defined user experiences and effectiveness of the clock.

While dealing with the development process, several activities focused on the main points. First, we made the system architecture, which consists of the DP, control unit, and ASM chart, and we modeled and tested the circuits on the logisim evaluation to show the control logic and the state transitions. Then, we wrote separate Verilog modules, for example, the 7-segment display driver, the second-minute counter, and the clock divider. We merged these modules and thoroughly inspected their performance based on comprehensive simulations. Lastly, we validated the design to the BASYS3 board by testing it practically to meet all its requirements and to operate reliably. This collaborative effort underscored the importance of teamwork in engineering projects, where diverse perspectives and skill sets can significantly enhance the quality and efficiency of the final product. Moreover, the digital alarm clock project emphasized the practical application of theoretical knowledge. Concepts such as finite state machines, clock division, and digital counters, which are often abstract in a classroom setting, were brought to life through hands-on implementation. This practical

experience is invaluable for solidifying understanding and preparing for more complex projects in the future.

Contributions of each member

Marwan Abudaif: was responsible for designing and testing the system using Logisim. He created the initial Logisim circuits to simulate the digital alarm clock's logic and verify the design before hardware implementation.

Mazin Bersy: focused on developing the datapath and control unit. He implemented the counters for seconds, minutes, and hours, and integrated the 7-segment display driver module.

Yousef Sayed: created the ASM (Algorithmic State Machine) chart. He detailed the state transitions and control logic required for the clock and alarm functionalities, ensuring a clear and structured design. Contributed in developing the report.

Amr Eid: contributed in the ASM , Created took charge of developing the counter modules for tracking seconds, minutes, and hours within the digital alarm clock. Contributed in developing the report.

All Members
All members collaboratively worked on the Vivado implementation, coding, testing, and debugging to ensure the digital alarm clock functioned correctly on the BASYS3 board.

System Design

Datapath and Control Unit.

The system design for our digital alarm clock project revolves around two primary components: the two parts of the processor, namely, the data path and the control unit. They cooperate to form the fundamental functionality and usability features of the alarm clock.

Data Path

The data path is the part of the clock which both counts time and times; including displaying the time on the 7-segment display and controlling the alarm functionality. The key components of the data path include: The key components of the data path include:

Clock Divider: This section takes the clock of 100 MHz and divides it down to 1 Hz that are used to sequentially increase the seconds counter by one every second after the other.

Counters: The hours and minutes are tracked using a seconds-minutes-hours counter system. These counters increment based on the 1 Hz clock signal.

7-Segment Display Driver: This module converts the binary count from the counters into a format suitable for display on the 7-segment displays.

Alarm Registers: Separate registers for storing hours and minutes that are compared with the current time.

Multiplexers (MUX): These are used to select between different data inputs, such as current time versus alarm time, to display the appropriate values on the 7-segment display.

LED Indicators: LEDs are used to indicate the current mode (clock/alarm mode or adjust mode) and which parameter (hours or minutes) is being adjusted.

Push Button Detectors: They recognize button presses and debounce the signals so the user input is accurate.

This Data path design guarantees that the clock is able to exactly keep time, display the current time and trigger the alarm at the predetermined time.

Control Unit

The control unit of the alarm clock handles the entire operation and state changes. It determines the user inputs from the push buttons and initiates the commands between different blocks in a data path.

Mode Control: This switches between clock/alarm mode and adjust mode based on what buttons are input.

Parameter Selection: It selsects which parameter (hours or minutes) is currently being adjusted in adjust mode.

Increment/Decrement Control: Managing the increment and decrement operations for adjusting the current time and alarm time.

Alarm Triggering: Activating the alarm when the current time matches the alarm time and handling the alarm off action when a button is pressed.

ASM Chart

The chart of the ASM (Algorithmic State Machine) presents an image of control unit functioning. It describes the changing states and the corresponding actions required in response to different data and situations. The ASM chart for our digital alarm clock includes the following states and transitions:The ASM chart for our digital alarm clock includes the following states and transitions:

Current Clock Time: The initial state where the clock do its job by update time every second.

Alarm Triggered: When the current time meets the alarm time, the system prompts to blinking LEDs.

LED0 ON/OFF: Shows that there will be LED light on when there is trigger of alarm and LED light off otherwise.

Adjust Mode: The menu is entered when BTNC button is pressed, it makes it easy for the user to adjust the time and set the alarm time also.

Adjust Hours/Minutes: States for adjusting the hours and minutes, controlled by BTNL and BTNR for selection, and BTNU and BTND for incrementing and decrementing values.

This is the ASM chart for the alarm clock logic and if it is properly understood, it would assist in implementing the control unit.

Logisim Evaluation

The system design evaluation part of our system underwent a Logisim clock circuit construction which is a digital logic simulation software. The Logisim evaluation can help us to verify real operations and functions of our design before actual implementation by FPGA. The key components of the Logisim clock circuit included:The key components of the Logisim clock circuit included:

Hours and Minutes Counter: These registers like the Verilog will keep track of the time.

Multiplexing: Through the multiplexing of time, the sequencing of frame was done in such a way that the display switched between hours and minutes.

The Logisim evaluation was instrumental in identifying potential issues and verifying the correctness of our design logic.

Implementation

The implementation phase of our digital alarm clock project involved developing various modules in Verilog, which were then integrated to form a functional system. Every single module was designed to perform a specific task required for the functioning of the entire clock, from accurate timekeeping to displaying it on the 7-segment display and handling user interactions. In the following, we will provide an overview of the modules and their implementation.

Seven Segment Display Decoder with Enable

The module SevenSegDecWithEn consists of a 7-segment decoder and an enable signal, the purpose of which is to translate BCD data into displayable 7-segment segments. Moreover, it provides strategies for the enable signals to ensure that the right digit is displayed.

Seconds-Minutes-Hours Counter

The SecMinHourCounter module handles the counting of seconds, minutes, and hours. It ensures that the time is correctly incremented and wraps around at the appropriate values (60 seconds, 60 minutes, 24 hours).

Multiplexers:

4-to-1 Multiplexer (mux4to1)

The mux4to1, on the other hand, uses a 4-bit input to choose from among four 4-bit inputs when supplied with a 2-bit input selection signal. We combine a character multiplexer to control which digit (seconds, minutes, hours, or alarm) is output on our 7-segment display.

2-to-1 Multiplexer (mux2to1)

The mux2to1 module selects one of two 4-bits inputs according to bit length instead of 1-bit selection signal. This multiplexer is used in different parts of the system where a selection between two binary is input.

Modulo Counters:

Modulo-24 Counter (mod24counter)

The mod24counter module counts from 0 to 23 and then resets to 0. It is used for counting the hours in a 24-hour format.

Modulo-60 Counter (mod64counter)

The mod60counter module counts from 0 to 59 and then resets to 0. It is used for internal clock division and other timing operations.

Debouncer

The debouncer module filters out spurious signals from mechanical button presses.

Synchronizer

The synchronizer module aligns the signal from the debouncer with the system clock to prevent metastability.

Pushdown Detector

This module combines the debouncer and synchronizer with a rising edge detector to detect stable button presses.

Rising Edge Detector

Uses a filp flops to detect the rising edge of an input signal

Digital Clock Module

The digitalClock module is the top-level module that integrates all other modules to form the complete digital alarm clock. It takes in the clock signal, enable signal, and reset signal as inputs, and outputs the signals to drive the 7-segment displays.

Clock Divider

The clockDivider module is used to generate slower clock signals from the 100 MHz input clock. These slower clocks are used for various timing operations within the clock.

Binary Counter

The BinaryCounter module is a generic counter that can be configured for different bit widths and maximum counts. It is used for counting time and generating clock signals.

7-Segment Display Driver

The SevenSegDecWithEn module takes a BCD value and converts it to the appropriate signals to drive a 7-segment display. It also manages the enable signals to multiplex the display.

Mode Selection Logic

The mode selection logic handles switching between normal clock mode and adjust mode. It interprets button presses to allow the user to set the current time and alarm time.

Alarm Functionality

The alarm functionality compares the current time with the set alarm time. When the current time matches the alarm time, the alarm is triggered, and an LED indicator blinks to alert the user.

States of the Digital Clock System

Clock_mode

Purpose: This is the basic working mode of the clock where this time is persistently shown on the display.

Functionality
Clock keeps ticking and the numbers of the hours and minutes representing the current time are all displayed.
A monitoring of buttons takes as input changes to move to other states.
In case the alarm is still enabled and the current time is matching the alarm time, the system will be therefore changed to the Alarm_mode

2. Adjust_clockmin

Purpose: This situation is meant for depicting those tiny seconds that come with every clock time.
Functionality:
The user can change the minutes by pressing the buttons for the state "increment" or "decrement".
The Clock, clocks the timing mechanism itself, with the specific change happening in the minutes.
A button push debounce and synchronize make careful minute steps possible.

In seconds, the user is able to select the minute duration; from there, they may change the hours or they go back to the previous mode.

\

3. Adjust_clockhour

Purpose: This condition is applied for synchronized adjustments related to time of the clocks.

Functionality:

User is to make use of the plus and minus buttons for a similar effect in order to add or subtract an hour respectively.

The Clock signal is very much like the Adjust_clockmin's. It is disabled when the hours change. "Time system" makes sure 24-hour cycle with hour value wrapping comes through.

The user can shift the time mode either from adjusting the hours to adjusting the minutes or through normal working state via state transition mechanism.

4. Adjust_alarmmin

Purpose: As on an alarm clock, this state would serve to specify the minutes of time setting.

5. Adjust_alarmhour
Purpose: This service is responsible for setting the time of alarm.
Functionality:
This allows the user to use the set buttons to change alarm hours.
After setting the alarm hours, the user can finalize the alarm setting or transition back to the normal clock display.

6. Alarm_mode
Purpose: This state is activated when the alarm is triggered.
Functionality:
When the current time matches the preset alarm time, the system transitions to this state to activate the alarm.
The alarm mechanism is engaged, which could include sounding a buzzer, flashing a light, or another alert method.
The user can deactivate the alarm, returning the system to the normal clock mode.

Implementation Issues

In the process of working our way through an issue of the digital alarm clock project we faced some challenges and problems. Adding to the fundamental challenges of the project correct working and reliability of the final solution was crucial.

1. Debouncing Button Inputs
Issue: Button presses often lead to noise, which after all means a double or more press registrations. This can cause unintended variance of some states and of the inputs of the users.

2. Synchronizing Asynchronous Inputs

Issue: Asynchronous inputs, such as button presses, can cause metastability when interfaced with synchronous digital circuits. This can lead to unpredictable behavior in the system.

3. Edge Detection

Issue: Detecting the rising edge of signals (e.g., button presses) is crucial for accurately capturing user inputs and initiating state transitions.

4. Clock Divider Design

Issue: The system required different clock frequencies during multiple processes, such as a fast clock for debouncing and a slower clock for timekeeping tasks.

5. State Transition Handling

Issue: Ensuring seamless and correct transitions between states for example adjusting time, setting alarm is challenging, especially when handling multiple button inputs.

6. 24-Hour Format Management

Issue: we had countless attempts as we could not decide about the 24-hour format for both clock and alarm.

7. Alarm Management

Issue: The system should be properly configured so as to detect, when the system's precise time bears the same value as with the alarm clock time, therefore switch to the alarm state.