



## 8086 Stopwatch

# Table of Contents

1. Introduction
2. Overview of the 8086 Microprocessor
3. Program Description
4. Functionality
5. User Commands
6. Technical Implementation
7. Conclusion

## 1. Introduction

The 8086 Stopwatch program is an assembly language application designed to demonstrate the capabilities of the 8086 microprocessor. The primary function of this program is to track elapsed time in a format that is user-friendly and efficient. By leveraging the unique features of the x86 architecture, the program provides an interactive console interface for time tracking. This report outlines the program's design, functionality, and implementation details.

## 2. Overview of the 8086 Microprocessor

The Intel 8086 microprocessor, introduced in 1978, marked a significant advancement in computing technology. It was one of the first processors to implement a 16-bit architecture, allowing for increased data processing capabilities compared to its predecessors. Key features of the 8086 include:

- **16-bit Data Bus:** Enables the processor to handle 16 bits of data simultaneously, improving performance.
- **Segmented Memory Architecture:** Allows the processor to access a larger memory space than its immediate addressable range, utilizing segment registers to manage different memory segments.
- **BIOS Interrupts:** Facilitates direct communication with hardware and system services, enabling tasks such as input/output operations.

The 8086 architecture laid the foundation for future x86 processors and continues to influence modern computing.

### 3. Program Description

The 8086 Stopwatch program is implemented in assembly language, utilizing BIOS interrupts for user interaction and display output. The program counts elapsed time in hours, minutes, and seconds, and provides functionality to start, stop, and reset the timer based on user input. The design aims to be efficient and straightforward, emphasizing clarity and ease of use.

#### #Key Features:

- **Real-time Clock Functionality:** The program updates the displayed time every second.
- **User Control:** Users can manage the timer through simple keyboard commands.
- **Visual Output:** The time is displayed in a standard `HH:MM:SS` format for easy readability.

### 4. Functionality

The stopwatch program operates in a loop, continuously updating the time until the user opts to stop or exit the program. Below are the primary functionalities of the program:

- **Time Display:** Shows the current time in the format `HH:MM:SS`.
  - **Time Increment:** Every second, the program increments the seconds counter, with appropriate handling for minutes and hours overflow.
  - **Error Handling:** The program includes a custom divide error handler to prevent crashes due to division by zero.
- **Time Management Logic:**
- The seconds counter is incremented until it reaches 60, at which point it resets to 0 and increments the minutes counter.
  - The same logic applies for minutes and hours, allowing the timer to run continuously.

## 5. User Commands

The program features an interactive command interface, allowing users to control the stopwatch effectively. The following commands are supported:

- **S (Start):** Initiates the timer. The program begins to increment and display the time.
- **E (Stop):** Pauses the timer. The current time is held until the user decides to restart it.
- **R (Reset):** Resets the timer to `00:00:00`, clearing all previous time data.
- **ESC:** Exits the program gracefully, ensuring that any resources are released properly.

### User Interaction:

Upon starting the program, the user is presented with a brief instruction set displayed on the screen. This ensures clarity in operations and enhances the user experience.

## 6. Technical Implementation

The implementation of the 8086 Stopwatch is structured around the following components:

### Data Segment:

- Variables for hours, minutes, and seconds are declared, along with message strings for user interaction and display.

### Code Segment:

- The program initializes the data segment and sets up an interrupt handler for divide errors.
- A main loop is established to handle user input and update the timer display accordingly.

### Key Procedures:

- **Update Time** : Handles the logic for incrementing seconds, minutes, and hours, ensuring correct formatting and overflow management.
- **Convert BCD** : Converts binary data to Binary-Coded Decimal (BCD) for display purposes
- **Print String** : Utilizes DOS interrupt calls to output strings to the console.

### Interrupt Handling:

The program defines a custom interrupt handler for divide errors, ensuring that any unexpected division operations do not crash the application. This enhances the program's stability and reliability.

## 7. Conclusion

The 8086 Stopwatch program serves as an educational tool for understanding assembly language programming and the functionality of the 8086 microprocessor. By implementing a straightforward timer application, users gain insight into real-time processing and user interaction within a console environment. The project not only demonstrates practical programming skills but also reinforces the concepts of data handling and interrupt management. Overall, the program exemplifies the capabilities of the 8086 architecture in a user-friendly format.

**Names:**

جوزيف جورج  
عبدالعليم رمضان  
زياد سعد  
زياد جمال  
منه الله احمد  
محمد اشرف