

燕山大学课程设计报告

课程名称：汇编语言

题目：电子时钟音乐发生器

摘要

本课程设计基于唐都科教仪器，设计并实现了一款运用定时计数器 8254、并行输入输出接口 8255 和中断控制器 8259，设计出一个功能全面且实用的电子时钟音乐发声器。

系统不仅能够通过七段数码管实时显示当前时间，还具备丰富的功能，包括开始时间的设置、闹钟时间的定时、闹钟的删除和暂停、秒表计时、整点报时以及播放音乐等多种实用功能。

设计过程中，使用者通过并行输入输出接口 8255 与系统进行交互，方便地输入和修改闹钟时间，并利用该接口控制秒表的启动、暂停和重置等功能。电子闹钟通过中断控制器 8259 触发相应的中断，进而激活蜂鸣器播放预设的音乐。定时计数器 8254 生成精确的时钟信号

通过系统调试与测试验证，电子时钟音乐发声器可以长时间正常运行，在错误输入（时、分、秒输入>59）可进行有效处理，全面达成了课程设计的目标要求，提高了对电子技术的理解和应用能力，增强了实际动手能力。

关键词：汇编语言；电子时钟音乐发声器；系统交互与功能控制；

目录

摘要	1
1 前言	1
1.1 项目背景	1
1.2 目的的意义	2
1. 深化对硬件编程与中断控制机制的理解	2
2. 强化对系统总线与接口通信的整体认知	2
3. 提升将应用功能映射为硬件驱动与实时程序的能力	2
4. 培养基于硬件的系统集成与调试能力	2
2 主要任务	4
2.1 项目概述	4
1. 时钟设置及显示	4
2. 实时时钟功能	4
3. 秒表功能	5
4. 闹钟功能	5
5. 整点报时功能	5
6. 显示功能	5
7. 音乐播放功能	6
2.2 需求分析	6
2.2.1 功能需求	6
2.2.2 接口需求	7
3 总体设计	9
3.1 功能设计	9
3.1.1 硬件资源层	9
3.1.2 驱动控制层	10
3.1.3 功能模块层	10
3.2 界面设计	11
3.2.1 显示界面设计	11
3.2.2 交互控制界面设计	12
3.3 硬件设计	12
3.3.1 系统总线接口设计	12
3.3.2 并行 I/O 接口电路 (8255)	14
3.3.3 定时与发声电路 (8254)	14
3.3.4 中断控制电路(8259)	15
4 详细设计	16
4.1 整体思路	16
4.2 软件设计	16
4.2.1 程序总体结构设计	17
4.2.2 电子时钟显示模块	19
4.2.3 闹钟响铃	20
4.2.4 整点报时	22

4.2.5 设置时间	22
4.2.6 暂停时间	23
4.2.7 删除闹钟	24
4.2.8 秒表计时	25
5 遇到的问题与解决方法	27
5.1 数码管时间显示问题	27
5.2 键盘去抖问题	27
6 总结	29
参考文献	30

1 前言

本课程设计围绕汇编语言程序设计展开，要求在给定的教学仪器中，完成一个具有开始时间的设置、闹钟时间的定时、闹钟的删除和暂停、秒表计时、整点报时以及播放音乐等多种实用功能的电子时钟音乐发声器。在本节中，将对汇编语言与接口技术课程设计的项目背景和目的意义作简要引入。

1.1 项目背景

汇编语言与接口技术是计算机硬件相关课程中的关键应用与实践课程，其主要研究对象是计算机指令系统在软件层面的直接编程方法以及硬件与外部设备间的通信控制机制，包括 CPU 寄存器、内存寻址方式、中断系统、并行/串行接口、定时计数器等核心内容。该课程关注即从汇编指令到硬件信号、从软件逻辑到外部设备响应的完整实现过程。

在汇编语言与接口技术教学中，CPU 指令系统、外设接口与中断控制逻辑通常以芯片引脚图进行讲解，虽然能够帮助学生理解基本概念，但对于“应用程序如何通过汇编指令控制硬件接口实现复杂交互”这一问题，仍然较为抽象。为弥补纯理论教学的不足，唐都科教仪器通过可编程接口芯片，为学生提供了一个可编程、可交互、可观测的接口控制环境，使得抽象的硬件编程概念能够落实到具体的芯片初始化、端口读写和中断服务程序编写之中。

在前续实验中，我们已分别完成了汇编程序设计实验、8254 定时计数器基础应用实验以及 8255 并行输入输出接口简单控制实验，初步掌握了汇编语言编程、硬件定时控制以及端口读写的基本原理与方法。然而，这些实验多以单一芯片的独立编程验证为主，功能相对简单，尚未充分体现一个完整的计算机系统如何通过集成化的汇编程序，综合调度中断、通信与外部设备，以实现复杂实时交互任务的协同工作机制。

本次课程设计中，我组选择实现一个多功能电子时钟音乐发声器作为核心任务。综合运用汇编语言、定时/计数器（8254）、并行接口（8255）及中断控制器（8259）完成的综合性设计。其实现过程涵盖了实时时钟计算、键盘输入扫描、数码管动态显示、中断触发以及蜂鸣器音频控制等多个关键环节。通过在实验仪器上完成从时间设置、闹钟管理到音乐播放等一系列功能的设计与调试，不仅可以全面验证系统软硬件协同、多芯片编程与中断调度逻辑的正确性，也有助于加

深对计算机系统如何通过并行接口与外部设备进行实时交互，从而达到从单一芯片实验向综合系统设计跨越的教学目的。

1.2 目的意义

本实验的主要目的在于综合运用《汇编语言与接口技术》课程中所学的理论知识，完成一个集成化硬件接口系统的设计、编程与功能验证，并通过电子时钟音乐发声器这一具体应用，加深对计算机系统软硬件协同与实时控制过程的理解。其意义主要体现在以下几个方面。

1. 深化对硬件编程与中断控制机制的理解

通过实现时间管理、闹钟触发、音乐播放等完整功能，可以系统地理解如何通过汇编指令直接操控 8255、8254、8259 等可编程接口芯片。实验不仅关注功能实现，更强调端口读写、定时器配置、中断服务程序如何协同工作，从而加深对底层硬件控制思想的认识。

2. 强化对系统总线与接口通信的整体认知

在系统运行过程中，控制命令、时间数据、状态信号需要在 CPU、并行接口、定时计数器、中断控制器及外部设备（数码管、键盘、蜂鸣器）之间不断交互。通过对端口地址译码、定时器工作模式、中断优先级管理的反复调试与观察，可以更加直观地理解各接口芯片在系统中的角色以及它们之间的协同通信机制，从而建立起完整的外设接口与系统集成概念。

3. 提升将应用功能映射为硬件驱动与实时程序的能力

通过将电子时钟、闹钟、秒表、音乐播放等综合功能，分解为基于定时中断的时钟基准、键盘扫描输入、数码管动态显示、蜂鸣器频率控制等可编程硬件操作，有助于理解复杂应用在底层接口驱动中的实现方式。该过程要求对任务调度与中断响应进行合理设计，从而体现“通过汇编语言驾驭硬件”的思想，提升从功能需求到接口编程的抽象与实践能力。

4. 培养基于硬件的系统集成与调试能力

本综合实验过程中需要综合考虑芯片初始化、中断向量管理、实时性保障、人机交互逻辑等多个方面，有助于培养严谨的硬件编程习惯和调试能力。

综上所述，本实验不仅加深了对汇编语言与接口技术核心内容的理解，也为后续学习奠定了坚实的实践基础。通过亲身参与一个完整的接口控制系统从设计、编程到调试运行的完整过程，有助于全面理解计算机系统与外部世界进行实时交互的设计思想与实现机制。

2 主要任务

本项目围绕复杂模型机的整体功能设计与软硬件协同实现展开，基于 8259A 中断控制器、8254 定时器/计数器以及 8255 并行接口芯片，构建一个具备实时时钟、秒表、闹钟、整点报时、音乐播放及人机交互功能的综合电子闹钟系统。

2.1 项目概述

本项目基于 8259A 中断控制器、8254 定时器/计数器、8255 并行接口芯片构建，设计并实现支持电子闹钟，核心功能包括：

1. 时钟设置及显示

键盘输入：6 个七段数码管从左到右分别表示时间的时、分、秒，小键盘从上向下，从左到右，第一排按键分别表示数字 0、1、2、3；第二排按键分别表示 4、5、6、7；第三排按键分别表示 8、9、A、B；第四排按键表示数字 C、D、E、F。

KK1 是设置初始时间的按键。默认的初始化时间为 23:59:59，触按 KK1 后过小键盘进行数字输入，6 个七段数码管从左到右分别表示时间的时、分、秒，输入 6 个数，输入完毕即设置成功。

K7 是设置闹钟的时、分，以及选择曲目按键。上下拨动 K7 一次，进入闹钟设置，6 个七段数码管从左到右前四个分别表示时、分，第五个和第六个数码管表示闹钟的响铃曲目。

K6 是暂停时间，向上拨动一次 K6，暂停计时，拨回 K6，继续计时。

K5 是删除闹钟，上下拨动 K5 一次，删除已经设置的闹钟。

K4 是秒表计时，向上拨动一次 K4，开始秒表计时，拨回 K4，结束秒表计时。

2. 实时时钟功能

时间设置：通过矩阵键盘手动输入 6 位数字（时分秒各 2 位），输入后自动校验合法性（时 \leq 23、分 \leq 59、秒 \leq 59），非法输入则提示重新输入，设置成功后数码管同步刷新显示新时间。时间基准：利用 8254 定时器工作于模式 2 或 3，

计时控制：支持手动暂停/继续计时，通过功能按键触发 8255 端口检测，暂停时计时数据冻结，继续后从冻结时刻恢复累加，暂停状态下数码管保持当前时间显示。

3. 秒表功能

通过独立功能按键实现秒表启动、暂停、重置功能；启动后数码管切换显示秒表时间，暂停时保持当前计时值，再次启动从暂停值续跑；重置按键触发后秒表数据清零并恢复显示实时时钟。

显示切换：秒表工作时数码管优先显示秒表时间，停止后自动切换回实时时钟显示。

4. 闹钟功能

闹钟设置：支持通过键盘输入 2 位小时、2 位分钟（秒位固定为 00）设置闹钟时间，同时选择 1-4 号预设曲目（柯南主题曲、两只老虎、友谊地久天长、一剪梅），输入后校验时间合法性（时 \leq 23、分 \leq 59、曲目编号 \leq 4），非法则重新输入。

播放控制：闹钟播放时支持手动按键停止（通过键盘输入触发停止标志），停止后恢复实时时钟显示；

支持闹钟删除功能，通过功能按键 K5 清除当前闹钟设置，删除后闹钟状态置为无效。

5. 整点报时功能

通过 8254 定时器生成固定频率的音频信号，驱动扬声器播放 2 秒时长的报时提示音，音量与闹钟播放一致。

自动停止：报时持续 2 秒后，系统自动关闭音频输出，恢复正常计时状态，报时过程中不影响其他功能操作（如键盘输入、时间查看）。

6. 显示功能

显示内容：默认显示实时时钟（格式 HH:MM:SS, ）；秒表启动后切换为秒表时间（格式 MM:SS:XX, ）；按键操作时（如设置时间、选择曲目），同步显示输入的数字或功能标识。

7. 音乐播放功能

曲目存储：系统预设 4 首经典曲目，每首曲目对应独立的频率表（WORD 类型，存储每个音符的频率值）和时间表（BYTE 类型，存储每个音符的播放时长），频率范围 248Hz-1248Hz，适配 8254 定时器输出能力。

播放逻辑：根据选中的曲目编号，索引对应的频率表和时间表，通过 8254 定时器 B 生成对应频率的方波信号，驱动扬声器播放；每个音符按时间表设定的时长播放，播放完成后自动切换至下一个音符，直至频率表末尾（频率值为 0）结束。

音频适配：播放过程中保持频率信号稳定，无杂音；支持多曲目切换，闹钟触发时自动加载对应曲目，播放过程中可随时停止，停止后定时器恢复初始配置。

2.2 需求分析

2.2.1 功能需求

1. 实时时钟功能需求

系统应能够实现连续、稳定的实时时钟计时功能，支持对小时、分钟和秒的自动累加。

2. 时间设置功能需求

支持用户通过键盘输入方式手动设置当前时间。时间设置完成并确认后，系统应立即更新内部时间数据，并同步刷新显示内容。

3. 秒表功能需求

支持独立的秒表计时功能，支持启动、暂停、继续和复位操作，

4. 闹钟功能需求

支持闹钟功能，允许用户设置指定的闹钟时间，并在达到设定时间时触发音频提示。闹钟时间设置应支持小时与分钟输入，系统需对输入数据进行合法性校验。

5. 整点报时功能需求

支持整点报时功能，在每个整点时刻自动发出提示音。报时音频持续时间应有限，并在播放结束后自动停止。

6. 显示需求

支持数码管作为主要显示设备，能够清晰显示当前实时时钟或秒表计时数据。显示内容应根据系统当前工作状态自动切换。

2.2.2 接口需求

1. 芯片接口需求

(1) 8259A 中断控制器

中断响应：支持 IR6 中断（键盘时间设置触发）、IR7 中断（8254 生成的 20ms 定时中断）。

(2) 8254 定时器/计数器

地址分配：控制寄存器地址 06C6H (CON8254)，计数器 A 地址 06C0H (A8254)、计数器 B 地址 06C2H (B8254)、计数器 C 地址 06C4H (C8254)。

功能接口：计数器 A 工作在方式 3（方波发生器），生成 20ms 定时中断，用于时钟/秒表的秒累加；计数器 B 工作在方式 3，根据曲目频率表输出对应方波信号，驱动音频播放；控制口支持模式 3 (36H)、模式 3 (76H) 等初始化命令，适配不同功能需求。

数据格式：支持 16 位计数初值输入（先低 8 位后高 8 位），计数初值由频率计算得出（初值=4240H/目标频率），确保频率输出精准。

(3) 8255 并行接口

地址分配：控制寄存器地址 0606H (CON8255)，端口 A 地址 0600H (A8255)、端口 B 地址 0602H (B8255)、端口 C 地址 0604H (C8255)。

端口功能：端口 A（输出）用于数码管位选信号输出（6 位数码管的选通控制）；端口 B（输出）用于数码管段码信号输出（8 段+小数点的段码驱动）；

端口 C（输入）用于键盘行信号检测（低 4 位接收矩阵键盘行数据），控制口初始化命令为 89H（端口 A/B 输出，端口 C 低 4 位输入）。

2. 外部设备接口需求

（1）输入设备接口

矩阵键盘接口：采用 4×4 矩阵键盘，行线连接 8255 端口 C 的低 4 位（PC0-PC3），列线连接 8255 端口 A 的低 4 位（PA0-PA3），支持 16 个按键（0-9 数字键、6 个功能键），接口需支持 5V 电平信号，按键闭合时输出低电平。

功能按键接口：独立功能按键（暂停、停止、重置等）可复用矩阵键盘的未定义按键，或通过端口 C 的高 4 位扩展输入，接口需兼容按键去抖处理的电平检测逻辑（低电平有效，持续检测时间 $\geq 20\text{ms}$ ）。

（2）输出设备接口

数码管接口：6 位共阴极数码管，连接端口 A 输出，段选端直接连接端口 B 输出，支持 8 段+小数点显示，工作电压 5V，。

音频输出接口：扬声器通过三极管放大电路连接 8254 计数器 B 的输出端（06C2H），接口需支持频率范围 248Hz-1248Hz 的方波信号输入，输出音量适中，无明显失真。

3. 数据接口需求

内部数据存储接口：系统数据区存储实时时间（hour/min/sec）、闹钟参数（Chour/Cmin/SELSONG）、秒表数据（Shour/Smin/Ssec）、控制标志（alarm_stop_flag/count）等，支持 8 位字节（BYTE）和 16 位字（WORD）数据的读写，地址分配在 DATA 段，偏移量固定，便于程序调用。

曲目数据接口：预设曲目对应的频率表（如 KNFR_LIST、TWTIG_LIST）和时间表（如 KNTI_LIST、TWTIG_TIME）以数组形式存储在 DATA 段，频率表为 WORD 类型（2 字节/个），时间表为 BYTE 类型（1 字节/个），曲目结束标识为频率值 0，支持通过曲目编号索引数组首地址，实现快速调用。

中断数据 接口：中断服务程序（MIR6/MIR7）与主程序间通过全局变量（如 buffer、count）传递数据，支持中断触发时的数据读写，确保时间设置、计时累加等数据交互准确，无冲突。

3 总体设计

本章主要对本实验中基于 8086 微处理器的多功能电子钟与音乐播放系统的整体设计方案进行说明。从功能结构、软硬件协同以及数据通路组织等角度，对系统的整体架构进行分析。通过对 TDX-PITE 实验箱硬件资源、汇编指令控制逻辑以及功能模块之间关系的系统化描述，明确各组成部分在整体运行中的作用和相互联系。

3.1 功能设计

本系统整体功能架构分为三层，从下至上依次为硬件资源层、驱动控制层、功能模块层，各层职责明确、协同工作，具体设计如下：

3.1.1 硬件资源层

基于 TDX-PITE 实验系统的硬件架构，主要利用以下核心芯片及资源：

1. 微处理器与寄存器

使用 **8086 CPU** 及其内部通用寄存器 (AX, BX, CX, DX) 进行数据运算与暂存；使用变址寄存器 (SI, DI) 和段寄存器 (DS, SS, CS) 实现对频率表、时间表及显示缓冲区的寻址与管理。

2. 可编程并行接口

使用 **8255** 作为系统核心 I/O 扩展，端口 A (PA) 用于控制数码管位选及键盘列扫描，端口 B (PB) 用于数码管段码输出，端口 C (PC) 用于读取键盘行信号及控制状态监测。

3. 可编程定时/计数器 8254

通道 0 (计数器 0) 用于产生系统时钟基准 (100Hz/50Hz)，为时钟走时提供中断源；通道 1 (计数器 1) 用于产生不同频率的方波信号，驱动扬声器播放音乐。

4. 可编程中断控制器 8259

管理外部中断请求，主要处理来自 8254 的定时中断（MIR7/MIR6），实现时钟的秒级计数、时间刷新以及键盘扫描的触发。

3.1.2 驱动控制层

驱动控制层是硬件资源与上层应用的接口，通过编写汇编子程序实现对硬件的底层操作，主要包括：

1. 初始化驱动

包括 INIT8255、INIT8254、INIT8259 子程序，用于配置各芯片的工作模式、控制字及中断向量表，确保系统上电后处于就绪状态。

2. 显示与扫描驱动

DISPLAY 子程序利用动态扫描技术，通过 8255 控制数码管轮流显示缓冲区数据；CHECK_STOP_KEY 子程序通过行列反转扫描法，实时检测矩阵键盘的按键状态。

3. 音频驱动

通过向 8254 写入不同的计数初值改变输出频率，配合 DELAY 延时子程序控制发音时长，实现音符的播放与节奏控制。

4. 中断服务程序

编写 MIR7 和 MIR6 中断服务程序，利用 IRET 指令和堆栈操作，实现计数的累加、时间的进位逻辑以及现场保护。

3.1.3 功能模块层

模块层是系统的核心业务单元，基于底层驱动实现用户可见的具体功能，分为四大模块：

1. 数字时钟模块

基于定时器中断实现“时-分-秒”的自动进位逻辑。包含 ADCLOCK（时钟进位）和 ADCLOCK2（秒表进位）子程序，支持 24 小时制走时，并能处理进位溢出。

2. 音乐播放模块

内置多首歌曲的频率表（Frequency List）和时间表（Time List）。通过查表算法读取音符数据，驱动蜂鸣器演奏多首曲目，并支持播放过程中的按键暂停/停止功能。

3. 闹钟与设置模块

C8255SETCLOCK 模块允许用户通过键盘输入设定闹钟时间及选择闹铃曲目。系统在后台实时比对当前时间与闹钟时间，匹配成功即触发响铃。

4. 秒表计时模块

C8255SCONSECOND 模块提供独立于主时钟的计时功能，支持清零、启动、暂停操作，通过键盘交互实现对短时间的精确测量。

3.2 界面设计

本系统采用“数码管显示+矩阵键盘输入”的人机交互模式。输出界面用于实时显示时间、秒表数值及设置参数，输入界面通过扫描矩阵键盘接收用户的控制指令与数据设定，通过控制端口（8255）协调数据的收发与状态切换。

3.2.1 显示界面设计

系统主要采用实验箱上的 6 位 LED 七段数码管作为状态显示终端，采用动态扫描技术实现视觉上的同时显示。

1. 显示布局

数码管从左至右依次显示时（Hour）、分（Minute）、秒（Second），两两一组，符合“HH MM SS”的常规时钟显示格式。

2. 显示原理

段选控制：通过 8255 的 B 端口（PB0~PB7）输出字形码（由 DTABLE 和 DPTABLE 查找表提供），控制数码管内部 a~g 及 dp 段的点亮，显示数字 0~9。

位选控制：通过 8255 的 A 端口（PA0~PA5）输出位选信号，利用轮流选通的方式（左移/右移位选码），依次点亮 6 个数码管。

动态刷新：利用人眼的视觉暂留效应，在 DISPLAY 子程序中通过快速循环扫描缓冲区（Buffer），实现无闪烁的实时时间显示。

3.2.2 交互控制界面设计

系统通过 4x4 矩阵键盘作为指令输入源，通过行列扫描法识别用户操作。

3.3 硬件设计

本系统的硬件电路基于 8086 系统总线架构，以核心接口芯片为控制枢纽，通过地址译码与信号复用技术连接外设。硬件电路主要由系统总线接口、并行 I/O 控制电路、定时/计数电路以及中断控制电路四部分组成。

3.3.1 系统总线接口设计

系统总线是 CPU 与各功能模块通信的物理通道，如图 3-1 所示，主要包括数据总线、地址总线和控制总线。

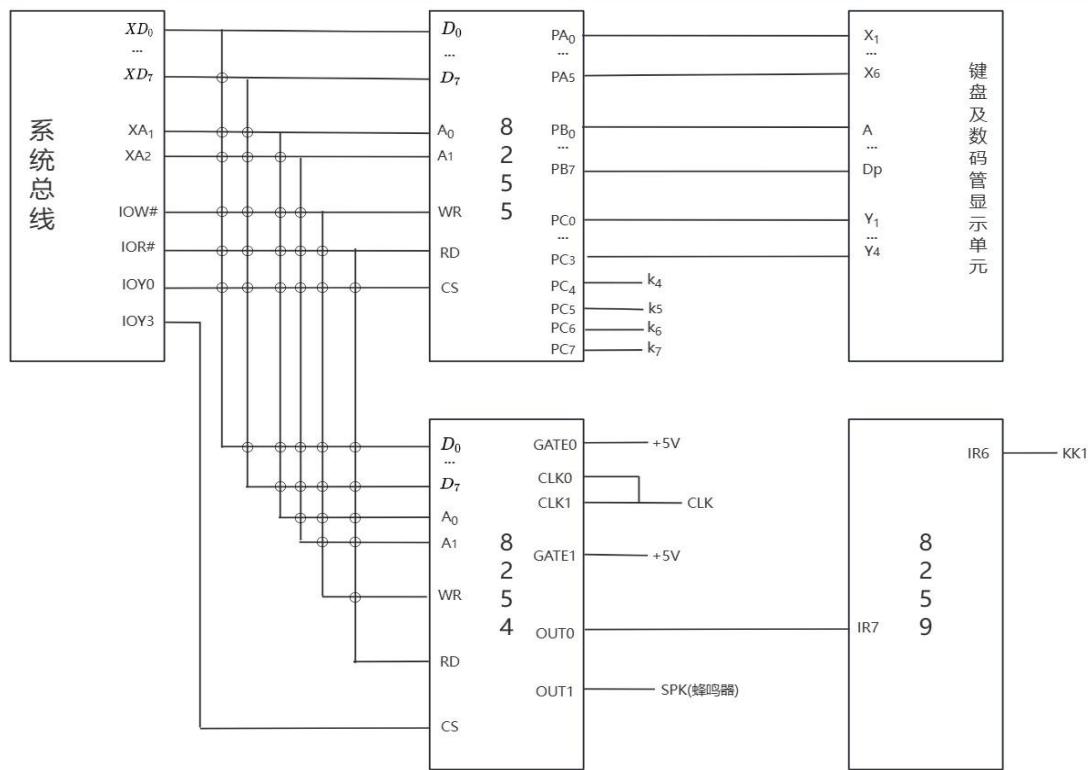


图 3-1 接线图

主要包括：

1. 数据传输

系统数据总线 $XD_0 \sim XD_7$ 直接连接至 8255、8254 的 $D_0 \sim D_7$ 引脚，实现 CPU 与接口芯片之间的 8 位并行数据交互。

2. 地址译码

系统采用部分译码方式进行片选控制：

8255 片选：地址译码输出信号 $IOY0$ 连接至 8255 的 CS 引脚，基址地址设定为 $0600H$ 。CPU 通过地址线 XA_1, XA_2 连接芯片的 $A_0, A_1$$ ，以选择 PA 、 PB 、 PC 端口及控制字寄存器。

8254 片选：地址译码输出信号 $IOY3$ 连接至 8254 的 CS 引脚，基址地址设定为 $06C0H$ 。同样通过 XA_1, XA_2 选择内部计数器通道。

3. 读写控制

系统的读信号 IOR 和写信号 IOW 分别连接至各芯片的 RD 和 WR 引脚，控制数据的流向。

3.3.2 并行 I/O 接口电路 (8255)

本系统使用一片 8255A 芯片作为人机交互的核心接口，负责驱动数码管显示和扫描矩阵键盘，端口分配如下：

1. 端口 A $PA_0 \sim PA_5$

复用输出口。连接至键盘/数码管模块的 X_1 ~ X_6 引脚。在显示周期，提供数码管的位选信号，控制 6 位显示位的亮灭；在键盘扫描周期，输出列扫描信号。

2. 端口 B $PB_0 \sim PB_7$

输出口。连接至数码管的 A ~ DP 引脚 (D_0 ~ D_7)。用于输出七段数码管的段码（字形码），控制显示具体的数字字符。

3. 端口 C $PC_0 \sim PC_7$

低 4 位 (PC_0~PC_3): 连接至键盘模块的 Y_1 ~ Y_4 引脚，用于读取矩阵键盘的行扫描信号，配合端口 A 实现按键检测。

高 4 位 (PC_4~PC_7): 连接至拨动开关 K_4 ~ K_7，用于读取外部配置状态或辅助控制信号。

3.3.3 定时与发声电路 (8254)

系统利用 8254 芯片提供精确的时基信号和音频频率，连接逻辑如下：

1. **计数器 0 (Counter 0):** 系统时钟源。

GATE0: 接 +5V，保持常开状态。

CLK0: 接系统时钟信号 CLK (如 1MHz 或特定频率)。

OUT0: 输出端连接至中断控制器 8259 的 IR7 引脚。计数器工作在方式 3 (方波发生器)，产生周期性的中断信号（例如每 10ms 或 20ms 一次），作为电子钟走时的时间基准。

2. 计数器 1 (Counter 1): 音乐发生器。

GATE1: 接 +5V，保持常开。

CLK1: 接系统时钟信号 CLK。

OUT1: 连接至扬声器/蜂鸣器 (SPK)。软件通过改变计数初值来改变输出方波的频率，从而驱动蜂鸣器发出不同音高的音符，实现音乐播放。

3.3.4 中断控制电路(8259)

8259 芯片负责管理系统的中断请求，本设计主要使用了两个外部中断源：

1. R7 (定时中断)连接 8254 的 \$OUT0\$ 引脚。当定时器计数溢出时触发中断，CPU 响应后进入 MIR7 中断服务程序，执行时钟计数累加和时间刷新逻辑。
2. IR6 (外部交互中断)连接至单脉冲发生器 KK1。用于手动产生中断信号，配合软件逻辑实现时间调整或特定模式的触发（在代码中对应 MIR6 中断处理，用于接收键盘输入设置时间）。

4 详细设计

4.1 整体思路

本电子闹钟系统基于 80x86 微处理器体系结构，综合运用定时计数器 8254、并行接口芯片 8255 以及中断控制器 8259A 等典型外设，采用汇编语言进行程序设计，实现电子时钟显示、闹钟定时、整点报时、时间设置、暂停计时及秒表计时等多种功能。系统整体设计遵循“硬件定时、中断驱动、软件控制、模块化实现”的思想，以保证系统运行的实时性、稳定性和可扩展性。

系统启动后，首先对各硬件模块进行初始化，包括设置 8259A 的中断向量表和中断工作方式，配置 8254 定时计数器的工作模式与初值，以及初始化 8255 并行接口的端口方向和控制字。同时，系统将初始时间设定为 23 时 59 分 59 秒，为后续时间计数与整点报时功能提供测试基础。硬件初始化完成后，系统进入主循环，开始执行电子闹钟的各项功能。

在时间产生与更新方面，系统以 8254 定时计数器作为时间基准源。通过对系统时钟进行分频，使计数器周期性地产生中断信号，并由 8259A 统一管理。当中断发生时，处理器进入相应的中断服务程序，对中断次数进行累计，当累计次数达到设定值后，实现秒数加一，并完成分钟和小时的进位处理，从而保证电子时钟的连续、准确运行。这种基于中断的计时方式避免了程序轮询带来的资源浪费，提高了系统的实时性。在显示与人机交互方面，系统通过 8255 并行接口与七段数码管和矩阵键盘相连接。数码管采用动态扫描方式进行显示，由程序依次选通各位数码管并输出对应段码，实现时、分、秒六位时间的稳定显示。矩阵键盘及功能开关用于完成时间设置、闹钟设置、暂停计时、秒表计时等操作，系统通过不断检测端口状态或响应中断请求，及时执行相应功能子程序。

在功能实现上，系统采用模块化的软件设计方法，将计时、显示、闹钟判断、整点报时、时间设置和秒表计时等功能分别封装为独立子程序，由主程序统一调度执行。各模块之间相互独立、逻辑清晰，既提高了程序的可读性和可维护性，也便于后续功能的扩展和修改。通过软硬件的紧密配合，最终实现了一个功能完善、运行稳定的电子闹钟系统。

4.2 软件设计

4.2.1 程序总体结构设计

本电子闹钟系统的软件部分采用 80x86 汇编语言编写，整体程序结构遵循“初始化—主循环—中断服务—功能子程序”的设计思想，通过中断机制与模块化程序结构相结合，实现系统各功能的协调运行。程序整体层次清晰，功能划分明确，便于调试和维护。

程序开始部分为系统初始化模块。在系统上电后，首先完成数据段、堆栈段和代码段的建立，并依次调用外设初始化子程序，对中断控制器 8259A、定时计数器 8254 以及并行接口芯片 8255 进行配置。通过设置中断向量表、初始化控制字和计数初值，使硬件工作于预定模式，为系统正常运行奠定基础。同时，程序对当前时间和闹钟时间等关键变量进行初值设定，确保系统进入可控状态。

初始化完成后，程序进入主程序循环。主循环不直接进行时间累加操作，而是负责调用显示子程序、功能检测子程序以及状态判断子程序。显示子程序通过 8255 向七段数码管输出数据，实现当前时间或秒表时间的动态显示；功能检测子程序负责读取开关和键盘状态，用于判断是否进入设置时间、设置闹钟、暂停计时、删除闹钟或秒表计时等功能模块。通过不断循环执行主程序，使系统能够实时响应用户操作。

系统的时间更新与关键事件处理主要依靠中断机制完成。定时计数器 8254 周期性地产生中断请求，经 8259A 处理后，CPU 进入相应的中断服务程序。在中断服务程序中，程序对中断次数进行累计，当达到设定阈值后完成秒、分、时的进位处理，实现电子时钟的准确计时。此外，部分功能如时间设置和秒表控制也通过中断方式触发，确保系统具有良好的实时性。

在软件实现上，各功能均封装为独立子程序，如时间累加、显示控制、闹钟判断、整点报时、音乐播放和秒表计时等。主程序通过调用这些子程序实现整体功能调度，各模块之间接口清晰、相互独立。该程序结构不仅保证了系统运行的稳定性，也增强了程序的可读性和可扩展性，符合汇编语言与接口技术课程设计的要求。程序实现整体功能调度，各模块之间接口清晰、相互独立。该程序结构不仅保证了系统运行的稳定性，也增强了程序的可读性和可扩展性，符合汇编语言与接口技

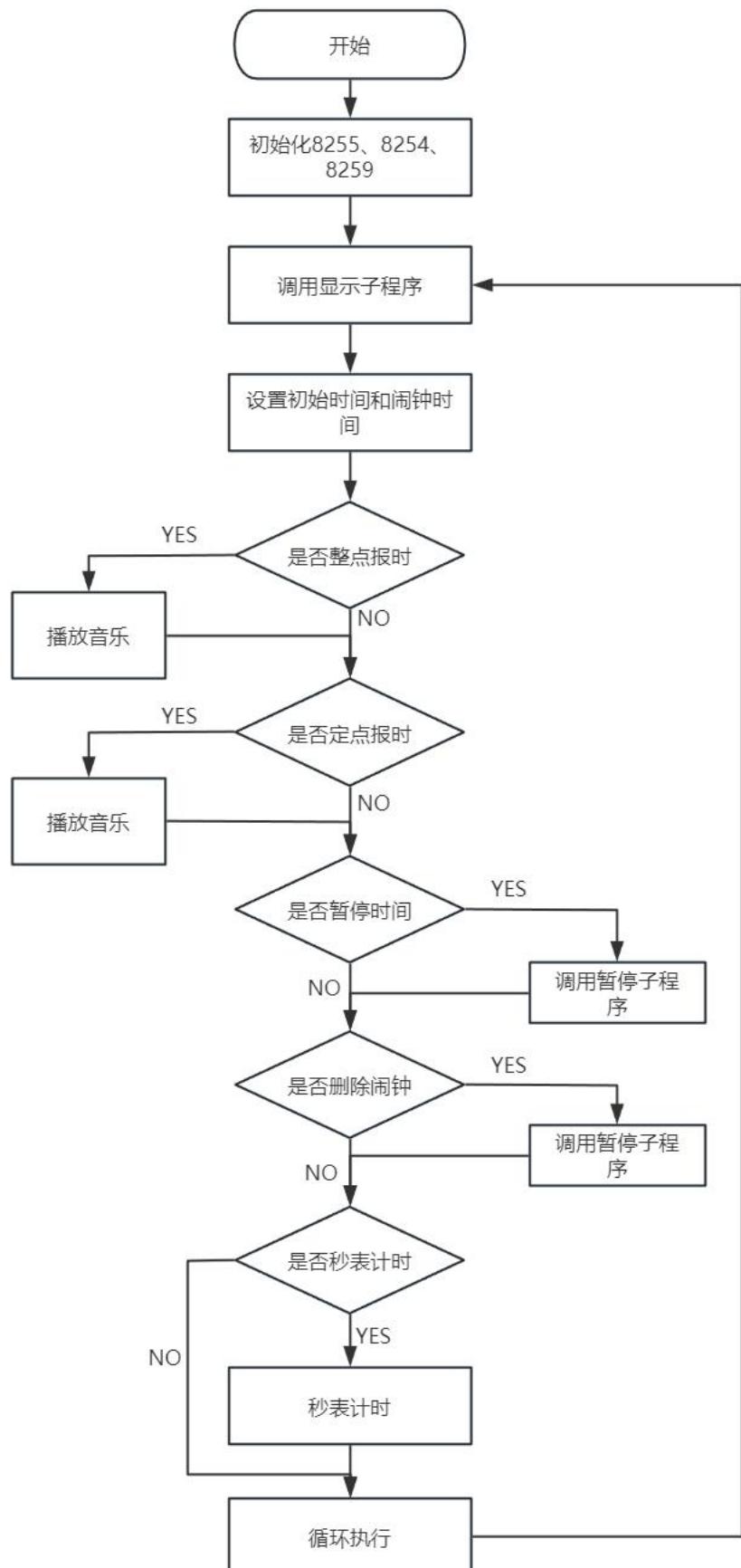


图 4-1 程序整体流程图

4.2.2 电子时钟显示模块

电子时钟显示主要完成计时，显示时间的功能。

首先计时部分，主要由 8259A、计数器 8254 完成。利用计数器 8254 分频功能，将系统时钟源 CLK 为 1MHZ 分为 50HZ，把计数器 0 的输出端 OUT0 接入主片 8259A 的中断引脚 IR7 上，实现每 20ms 计数器 0 给 80x86 系统一个中断。对中断进行计数，当中断次数达到 50 次时，秒数加 1，中断次数清零。

时间子程序用于累加时间。首先，检查秒钟是否大于 59 (CMP SEC, 3BH)。如果秒钟大于 59，表示超过了规定的秒数，将秒钟清零 (MOV SEC, 00H)，分钟加 1。然后，检查分钟是否大于 59 (CMP MIN, 3BH)。如果分钟大于 59，表示超过了规定的分钟数，将分钟清零 (MOV MIN, 00H)，小时加 1。再后，检查小时是否大于 23 (CMP HOUR, 17H)。如果小时大于 23，表示超过了规定的时间 (24 小时制)，那么将小时清零 (MOV HOUR, 00H)，即重置为零点。

中断执行过程和调用时间累加子程序的过程如下图所示：

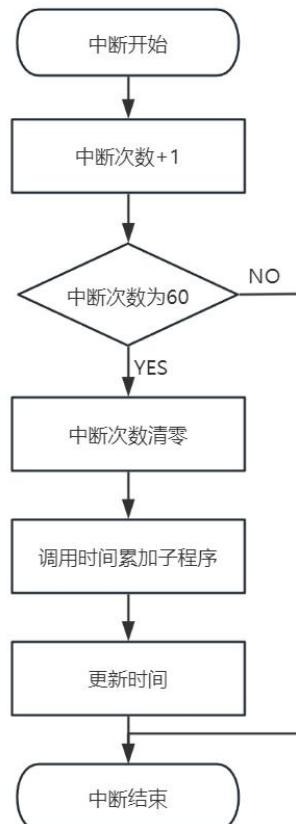


图 4-2 电子始终显示流程图

显示时间部分主要由并行接口 8255、七段数码管组成。系统利用 8255 并行接口输出功能向 6 个七段数码管输出数字，在数码管上显示时间，从左到右分别是时、分、秒。首先判断时间是否合法，接着将时、分、秒分离高位和低位，存储到 6 个连续的字节变量。然后用并行接口 8255 的 A 口输出数码管的选中信号，用 B 口输出数字对应的字形码。由于每次只能选中一个数码管输出字形码，因此循环执行 6 次就能完成完整 6 个时间数字的显示。每次输出完一个数字就进行延时，利用人的视觉暂留现象，让人感觉数码管是同时亮起。显示时间过程如下图所示：

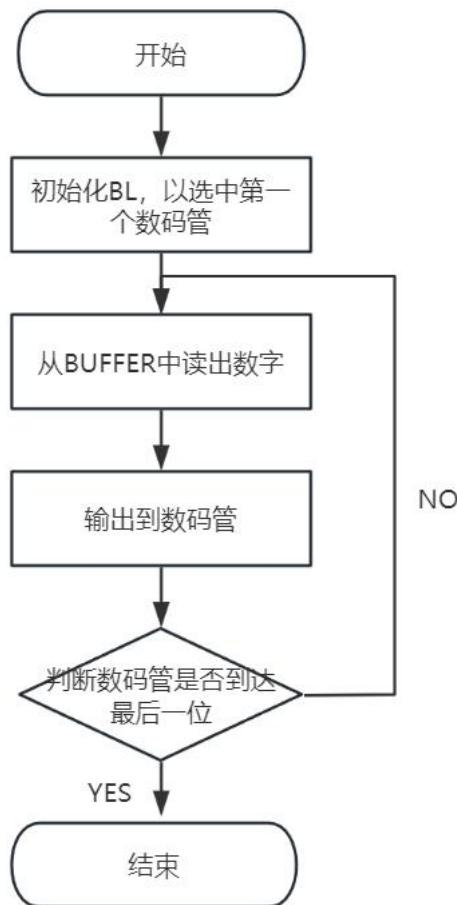


图 4-3 电子始终显示流程图

4.2.3 闹钟响铃

闹钟响铃功能的核心操作是通过矩阵键盘对闹铃时间进行精确设定，当电子时钟运行至预设的闹铃时刻时，系统将自动触发响铃提示。在设置过程中，用户可通过拨动 K7 开关一次完成状态切换，进入闹钟设置模式。设置内容依次包括小时的高位与低位、分钟的高位与低位，以及第五位用于选择闹钟提醒的歌曲曲

目。若输入的时间数值不符合规范（如小时超过 23 或分钟超过 59），系统将自动判定为无效设置，并提示用户重新输入，直至所有时间参数符合逻辑要求。整个设置流程清晰明确，用户可依照提示逐步完成，具体流程图如下图所示。

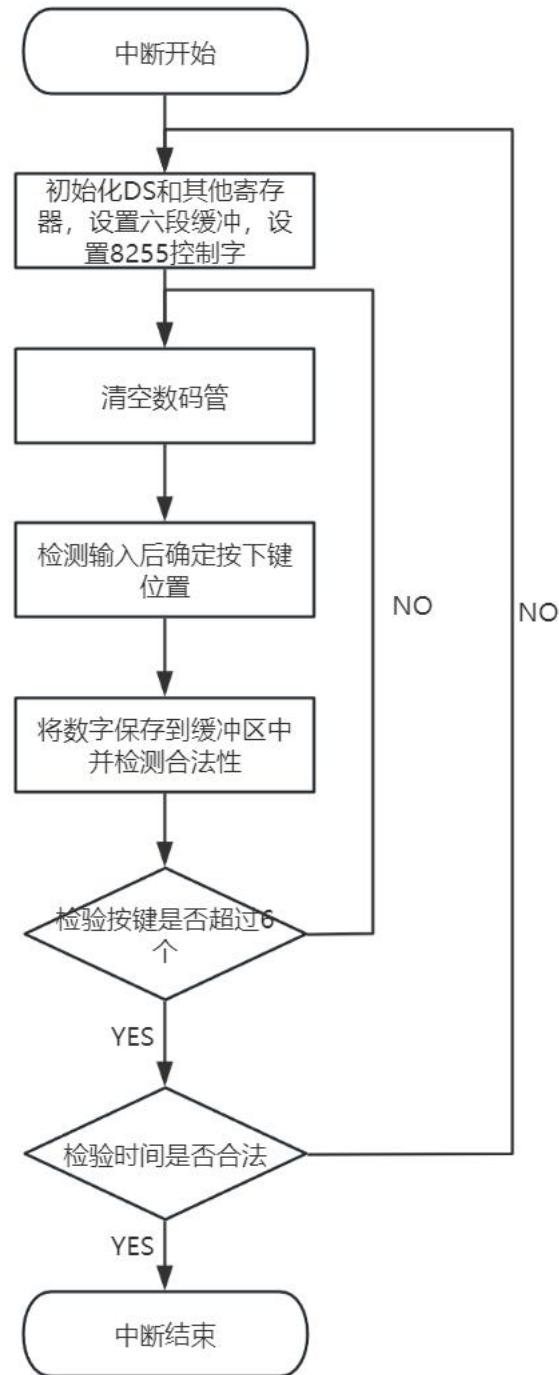


图 4-4 闹钟响铃流程图

响铃部分由定时器 8254、扬声器完成。首先判断当前时间是否为闹钟时刻，若为闹钟时刻则播放设置好的相应曲目。一个音符对应一个频率，将对应一个音

符频率的方波通到扬声器上，便能发出这个音符的声音。计算设置闹铃曲目的音乐频率表和时间表的偏移地址，控制 8254，设置计数初值，在扬声器上演奏相应的乐曲。在响铃时候可以按键结束闹钟。响铃过程如下图所示。

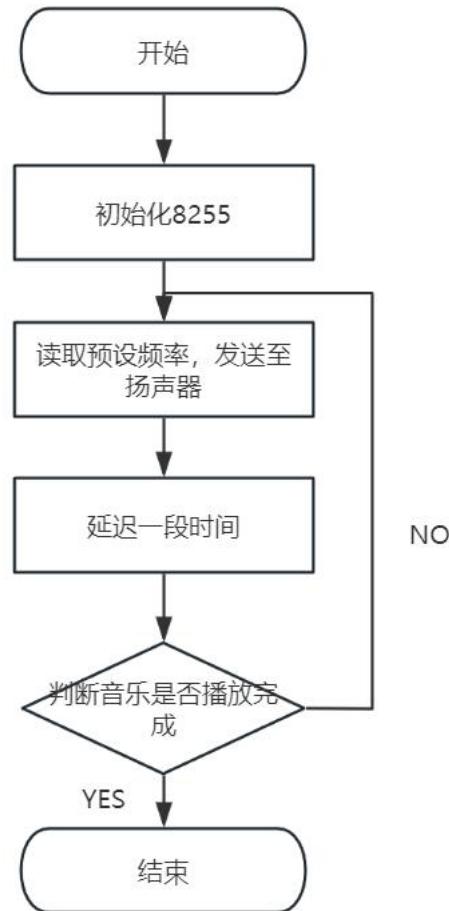


图 4-5 闹钟响铃流程图

4.2.4 整点报时

整点报时实现了时钟在运行到整点时刻扬声器能响 3 秒以达到提示作用。首先判断分钟和秒钟是否为 0，是则让 8254 计数器 1 输出特定频率的方波让扬声器响起，初始化 8254 计数器 1 工作于方式 3，给计数器 1 赋初值，当秒钟达到 3 时，扬声器停止，返回主程序。整点报时过程大致同上响铃过程。

4.2.5 设置时间

设置时间实现了在时钟运行的任何时刻能设置当前时刻。设置时间的信号主要由中断实现，按下按键开关 KK1，进入中断子程序设置时间。设置时间需要

从键盘上输入 6 个数，表示时分秒，确定键盘上按下键的位置并转化成数字。屏幕会显示设置的时分秒。矩阵键盘输完 6 个键后，中断子程序结束，返回主程序。设置时间的流程如下图所示。

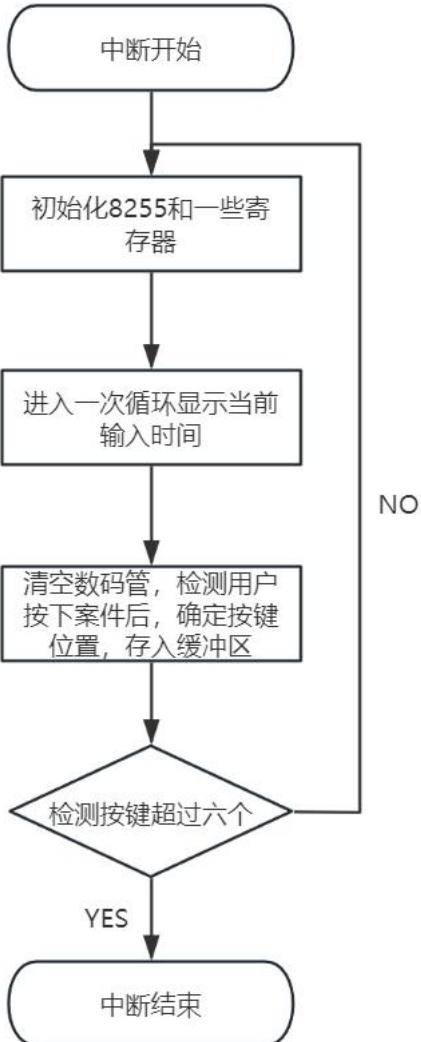


图 4-6 时间设置流程图

4.2.6 暂停时间

暂停时间功能的具体实现过程如下：当闹钟处于正常运行状态时，若用户主动拨动开关 K6，系统将立即检测到该操作并触发硬件中断机制，快速响应并跳转至预设的中断服务子程序。在此中断处理过程中，程序会暂停内部的计时操作和计数累加，确保时间数据暂时不再更新；同时，系统会保持当前时刻的持续显示输出，确保用户能够清晰、稳定地查看实时的时间状态，避免因计时暂停而导致

致显示混乱。该中断状态将一直维持，系统持续监听开关 K6 的状态变化，直到用户再次将开关拨回原始位置，此时中断条件解除，系统自动执行中断返回操作，结束当前中断子程序，随之完全退出中断处理模式，并无缝恢复至主程序中的正常执行流程，继续完成后续的计时与闹钟功能。整个暂停时间的完整流程还可参考下方附带图示以进行更详细的理解。

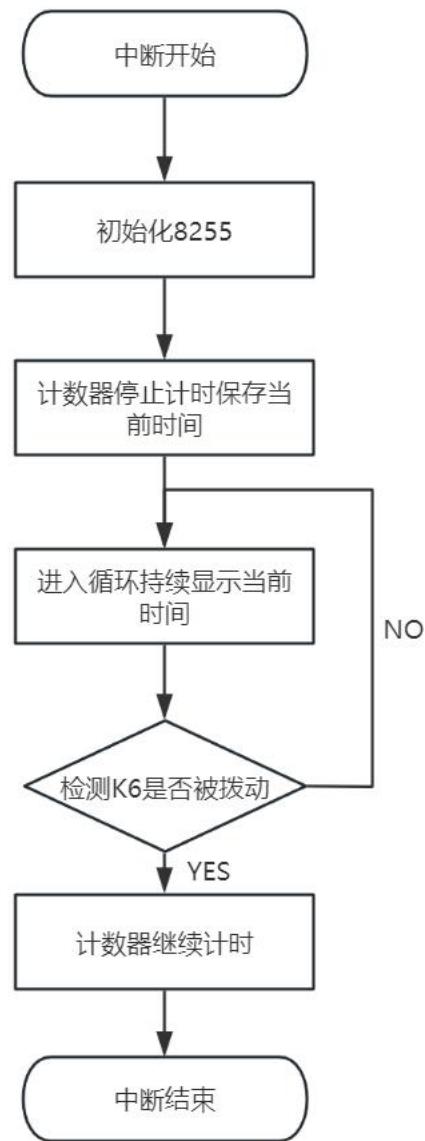


图 4-7 时间暂停流程图

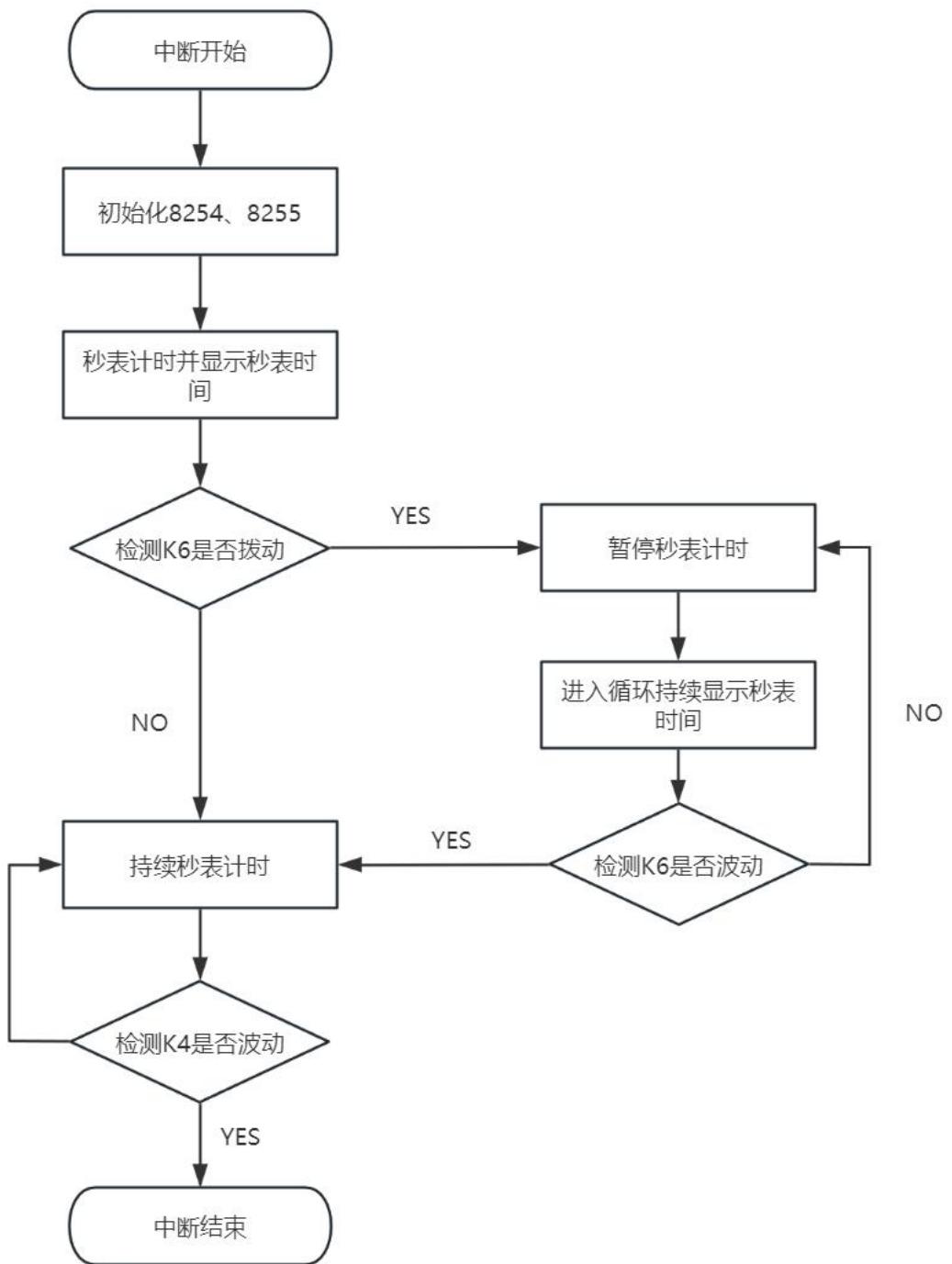
4.2.7 删除闹钟

删除闹钟的操作通过监测连接至 8255 并行接口芯片 C 端口的 K5 开关来实现。当拨动 K5 开关时，系统通过读取 8255 的 C 端口状态来检测这一动作。一

一旦检测到 K5 被激活，程序进入一个等待循环，这个循环将持续运行直到释放开关 K5，以此来确认删除意图。在等待循环期间，程序执行清除闹钟设置的操作，将存储闹钟时间的变量（Chour、Cmin、Csec）重置为 0，这表示闹钟时间已被清除，系统不再保留之前的闹钟设定。当用户释放 K5 开关后，系统检测到开关状态的变化，随即退出等待循环，完成闹钟的删除过程。此后，程序继续执行或返回主程序，闹钟功能将不再运行，直到重新设置闹钟时间。

4.2.8 秒表计时

秒表计时功能实现了在闹钟运行时，拨动开关 K4 时，进入中断子程序开始秒表计时，同时，时钟运行功能正常实现，数码管的六位依次显示当前秒表计时的时、分、秒的高位和低位，在秒表计时过程中，当拨动开关 K6 时，进入中断子程序暂停秒表计时，秒表停止计时，并持续显示当前秒表计时时间，直到再次拨回 K6，中断子程序结束。当再次拨动开关 K4 时，返回主程序，结束秒表计时，数码管显示时钟运行的时、分、秒。秒表计时的流程如下图所示。



5 遇到的问题与解决方法

5.1 数码管时间显示问题

问题描述：在数码管显示编程中，将存储的数字值直接转换为对应的段码字形并在正确位置添加小数点时遇到困难。

原因分析：直接转换逻辑复杂，需要同时处理数字分离、字形查表、小数点添加，逻辑嵌套导致程序结构混乱；带小数点转换易错：判断何时添加小数点的条件分支容易遗漏或错误，导致显示格式不正确。

解决方法：

1. 添加两个索引表分别为带小数点和不带小数点；
2. 创建格式化的转换函数；
3. 时间整体进行处理（hour->min->sec）十位从不带小数点的索引表（DTABLE）查，个位从带小数点的索引表（DPTABLE）查。

5.2 键盘去抖问题

问题描述：扫描编程中，检测按键按下并获取稳定键值时遇到困难，机械按键的抖动导致单次按压被误识别为多次输入。

原因分析：机械抖动固有特性，机械式按键在按下和释放时会产生 10-50ms 的物理抖动。

解决方法：设计 DALLY，每次检测输入进行跳转，DALLY 执行若干次循环避免抖动期。

```
BEGIN:  
    CALL DIS ; 显示当前缓冲区内容  
    CALL CLEAR ; 清屏  
    CALL CCSCAN ; 扫描是否有键按下  
    JNZ INK1 ; 有键按下则跳转
```

```
DALLY:  
    PUSH CX  
    MOV CX,0006H  
T1:    MOV AX,009FH  
T2:    DEC AX  
        JNZ T2  
        LOOP T1  
    POP CX  
    RET
```

JMP BEGIN ; 无键按下则继续扫描

INK1:

CALL DIS ; 显示

CALL DALLY ; 延时去抖

6 总结

本次汇编语言与接口技术课程设计围绕基于 8086 的多功能数字时钟系统设计与实现展开，通过团队协作完成了系统总体设计、模块划分、接口编程、指令编写、调试测试等全流程工作，成功实现了实时时钟显示、闹钟设置与提醒、秒表计时、整点报时及多曲目音乐播放等综合功能。。

在项目实施过程中，我们深入理解了 8086 微处理器的中断系统和工作原理，重点掌握了 8254 定时器、8255 并行接口、8259 中断控制器等外围芯片的编程控制方法，以及数码管动态扫描显示、矩阵键盘扫描输入、音乐频率合成等关键技术。通过对时钟管理、闹钟控制、音乐播放等多个功能模块的差异化设计，熟练运用了中断服务程序、端口 I/O 操作、数据转换处理等各类编程技术。同时，在解决数码管显示抖动、键盘去抖处理等问题过程中，培养了严谨的工程思维和硬件调试能力。

本次项目也存在一些可改进之处：例如可以改进防抖过程，原过程依赖固定的循环，具有一定的不稳定性，后续可以对该过程进行优化。

通过本次课程设计，我们深刻体会到了“硬件是基础，软件是灵魂”的设计理念，以及软硬件协同设计在嵌入式系统开发中的重要性。不同功能模块对硬件资源的差异化需求，促使我们深入思考如何合理分配中断资源、优化端口使用、平衡实时性与可靠性等技术挑战。这些实践经验为我们后续学习嵌入式系统开发奠定了坚实基础，也为我们未来从事硬件相关研发工作积累了宝贵经验。

参考文献

- [01] TDX-PITE_微机原理实验教程-学生版
- [02] 微机原理及接口技术试验教程（5.2 版）
- [03] 王爱英 杨蔚明. 计算机组成与结构[第五版]. 1 版. 清华大学出版社, 2024 年
- [04] 陈琦. 计算机组成原理实验教程[M]. 北京: 高等教育出版社, 2019: 137-167.