

项目 1 系统框架与芯片选型

【项目描述】

针对基于 STM8S 的步进电机运动控制系统总体需求,作为系统的第一个子项目,主要确认方案框架和功能设计等前期准备工作,该系统具体功能包括:

- (1) 步进电机是双闭环控制系统;
- (2) 功能按键具备转向和转速的设置;
- (3) LCD 液晶显示当前电机转速和转向。

【项目构思】

根据项目描述,确定项目需要使用的资源,内部资源有 I/O 端口和外部中断,外围模块有按键和 LCD1602 液晶模块。项目功能与章节知识对应关系如图 P1-1 所示。

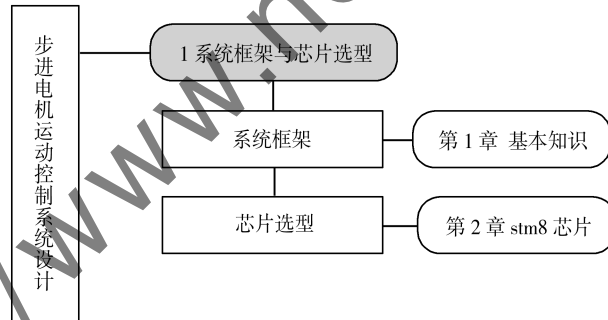


图 P1-1 项目功能与章节知识对应关系图

【项目设计】

根据项目描述和构思,从系统框架,器件选型和开发流程等方面进行项目设计。

1. 系统框架设计

系统框架如图 P1-2 所示。

2. 器件选型原则

MCU 是电子产品的核心部件,涉及硬件和软件,对于这个器件的选型需要非常慎重。需要综合考虑 MCU 要完成的功能,成本,交期,各类端口的数量,存储空间,可移植性,器件等级,功耗,电压等因素。

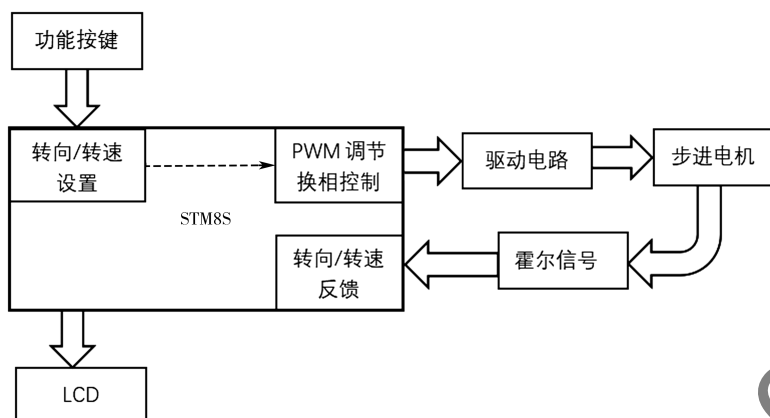


图 P1-2 系统框架

选型时需要考虑的问题点如下：

- (1) 应用场景和需求：软件、硬件和器件工程师共同确认产品需求、功能等情况；
- (2) 器件等级：包括器件等级，温度、功耗、安全等要求等；
- (3) 基本性能：包括处理器的类型、位数、时钟、内存、工作电压等基本参数；
- (4) 外设和接口：外设、接口，ADC/DAC、GPIO 口的种类和数量等；
- (5) 价格和品牌：对品牌和价格要求，供应链可靠性等；
- (6) 开发环境：开发环境和源代码、代码可移植性、技术支持状况等；
- (7) 安全性：根据需要选择有安全特性的处理器保证应用和数据安全。

3. 嵌入式开发流程

嵌入式软件开发的一般流程为需求分析、软件概要设计、软件详细设计、软件实现和软件测试。主要步骤如下：

- (1) 沟通准确的项目需求，明确开发任务及周边电路；
- (2) 根据需求进行 MCU 及关键器件选型，获取技术资料/软件代码/开发板/样品；
- (3) 借助 IDE(集成开发环境)：创建项目、配置引脚和外设、初始化系统时钟、初始化外设和编写主程；
- (4) 进行代码编译，确保代码符合语法规则；
- (5) 在样板调试之前，先用仿真软件进行系统仿真和调试；
- (6) 在软硬件设计完成后，在样板上进行系统联调，直到最后功能无误。

【项目实施】

参考项目设计，具体实施步骤如下。

1. 器件清单

根据系统设计方案和器件选型原则，列举了基于 STM8S 的步进电机控制系统的元器件清单，如表 P1-1 所示。

表 P1-1

器件清单

型号	说明	型号	说明
STM8S103F3P6	mcu 最小系统	A4988	驱动电路芯片
28BYJ-48	步进电机(内含霍尔传感器)	K1/K2 按键	功能按键:正转、反转
LCD1602	液晶显示	K3/K4 按键	功能按键:速度加、速度减

2. 原理图绘制和 PCB 回板

(1) 下载绘制软件:EDA 或 AD(Altium Designer)。

(2) 原理图绘制:创建工程文件,画布属性,电气工具选择,导线模式,总线分布,网络标签,标识符,网络端口,非链接标志,电压探针,引脚放置等过程。

(3) PCB 设计:准备好原理图库和 PCB 库,PCB 布局,PCB 布线,布线优化和丝印,DRC 检查和结构检查。

【项目运行】

1. 回板焊接

(1) 器件焊接:当 PCB 回板后,将购买的元器件进行焊接。

(2) 下载运行:将调试好的软件程序,下载进板子,确保程序可调,器件无误。

2. 观察效果

观察程序运行效果,如果与目标效果不同,修改代码或重新购买器件,直至达到目标效果。

第 1 章

基础知识

【主要知识内容】

单片机是一种高度集成的电子设备,它将计算机的多个关键组件集成在一个小型芯片上。目前,已广泛应用于各种嵌入式系统,如家用电器、汽车电子、工业控制等领域,因其体积小、成本低、功耗低而受到青睐。

本章主要介绍了计算机的基本概念,计算机硬件的基本组成,衡量计算机性能的关键指标,如处理速度、内存容量等。解释了嵌入式系统是什么,以及它与通用计算机系统的区别;探讨了嵌入式系统的发展历程和它所具有的独特特点,又根据应用领域或功能,对嵌入式系统进行了分类,以及描述了构成嵌入式系统的基本元素。最后,概述了 STM8 单片机的主要特点,STM8 单片机的内部结构,包括它的 CPU、存储器、外设等。通过本章的学习,会对单片机和嵌入式系统有一个基本的认识,为接下来学好其他章内容打好基础。

【学习目标】

- 理解计算机系统的基本概念和组成部分;
- 了解嵌入式系统的基本组成和应用;
- 了解 STM8S 单片机的特点,类型和基本结构。

【重点与难点】

- 重点:理解 STM8 单片机的基本组成和应用,嵌入式的基本概念和结构。
- 难点:STM8 单片机和嵌入式系统的区分。

【教学设计与实施方法】

本单元主要采用讲授教学法和练习教学法。讲授教学法通过教师课堂讲授实施,练习,教学法通过学生课后作业完成。

【实践环节设计】

本单元涉及计算机系统,STM8 单片机结构等内容。

【教学效果测评】

通过提问、课后习题、项目实训等方式进行效果测评。

1.1 计算机系统概述

计算机系统由两个核心部分构成:硬件和软件。

硬件,指的是计算机系统中物理存在的部分。它包括所有可见、可触摸的电子元件,以及各种光电机械设备,例如计算机的主机和外围设备等。

软件则是无形的,由人们预先编写并设计的,用以执行特定功能的程序和数据。这些软件可以存储在不同类型的存储介质上(如 RAM、ROM、磁带、磁盘、光盘,甚至纸带等),它们被加载到计算机的内存或辅助存储设备中。软件的发展不仅能够最大限度地发挥硬件的功能,提升工作效率,而且已经进化到可以部分模拟人类的思维活动。在计算机系统中,软件的地位和作用已经成为衡量计算机系统性能高低的一个关键指标。当然,计算机系统的性能是由软件和硬件功能的综合体现所决定的。

1.1.1 计算机的概念

在 1945 年,数学家约翰·冯·诺依曼(John von Neumann)在研究 EDVAC 计算机时,提出了一个革命性的概念——“存储程序”(stored-program)概念。基于这一概念构建的计算机被称为冯·诺依曼型计算机,其核心特点可以概括为以下几点:

(1)五大部分构成:运算器、存储器、控制器、输入设备和输出设备。

(2)指令和数据的统一存储:在这种架构中,指令和数据都以二进制形式平等地存储在存储器中,并且都可以通过地址来访问。

(3)二进制表示:指令和数据都用二进制数来表示,这简化了计算机内部的信息处理。

(4)指令的结构:每条指令由操作码和地址码两部分组成。操作码定义了指令的操作类型,而地址码则指定了操作数在存储器中的具体位置。

(5)指令的顺序执行,存储器中的指令通常是按顺序排列的,计算机按这一顺序逐一执行指令。尽管在大多数情况下指令是顺序执行的,但在特定条件下,可以根据计算结果或预设条件来改变执行顺序,实现程序的分支和循环。

(6)运算器为中心:在冯·诺依曼架构中,运算器是计算机的核心,输入设备和输出设备与存储器之间的数据传输都需要通过运算器来完成。

冯·诺依曼的这一概念对后来计算机的设计和开发产生了深远的影响,至今大多数计算机仍然采用这一架构或其变体。其基本结构如图 1-1 所示。

在现代计算机架构中,通常将负责算术运算和逻辑操作的运算器,与控制数据流的控制器集成在一块芯片上,称为中央处理器(Central Processor Unit,CPU),在某些情况下,也被称作微处理器(Microprocessor Unit,MPU)。为了进一步缩小系统体积、增强系统稳定性并降低制造成本,设计者们将输入/输出接口、时钟电路、存储器以及运算器和控制器等关键组件集成于单一芯片之中,形成了所谓的单片机(Microcontroller Unit,MCU)。单片机的设计理念在于,一个芯片便包含了一个完整计算机系统所需的所有必要组件。

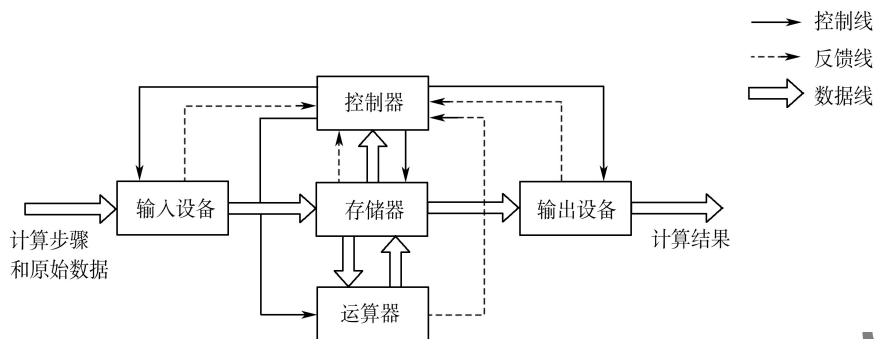


图 1-1 计算机系统的基本组成

针对多样化的应用程序,还会将具有不同功能的外设,例如定时器、中断控制器、模数(AD)与数模(DA)转换器、串行通信接口(例如 UART、SPI、I2C 或 CAN 等),甚至 LCD 显示驱动电路等均集成到一个芯片中,形成系列化产品,从而打造所谓的“嵌入式”单片机控制器(Embedded Microcontroller)。

1. 总线的概念

在计算机系统中,总线扮演着至关重要的角色。总线是连接各个组件并允许它们之间进行通信的共享通信路径。在早期的模拟电路设计中,组件之间的连接较为简单,通常采用串联方式。然而,在以微处理器为核心的现代计算机系统中,由于组件众多,若继续采用串联方式连接,将导致线路复杂且数量庞大。因此,现在普遍采用总线连接方式:所有组件的数据线和地址线都汇聚在一起,形成数据总线和地址总线,然后与 CPU 的数据和地址总线相连,构成并行结构。

为了避免通信冲突,要求在任何给定时刻只有一个设备能够与 CPU 进行通信,这需要通过控制线进行管理和选择,激活相应的芯片选择信号(如 \overline{CE} 或 \overline{CS})或输出允许信号 \overline{OE} 。

计算机系统内部的信息传输依赖于系统总线,它们可以根据传输的信息类型分为三种主要类别:数据总线、地址总线和控制总线。

(1) 数据总线(Data Bus)

数据总线负责在计算机的不同功能单元之间传输实际的数据信息。它是双向的,意味着数据可以在同一总线上来回传输。数据总线的宽度(即位)与计算机的字长和存储字长紧密相关,常见的宽度有 8 位、16 位或 32 位。数据总线的宽度是衡量系统性能的关键参数之一。例如,如果数据总线宽度为 8 位,而指令字长为 16 位,则 CPU 在取指令时需要两次访问主存储器才能获取完整的指令。

(2) 地址总线(Address Bus)

地址总线用于指定数据总线上数据的来源或目的地在主存储器中的地址,或者输入/输出设备(I/O)的地址。例如,要从存储器中读取数据,CPU 需要将该数据所在存储单元的地址通过地址总线发送出去。同样地,如果要通过 I/O 设备输出数据,CPU 不仅需要将该数据发送到数据总线上,还需要将该设备的地址发送到地址总线上。地址总线上的编码指明了 CPU 想要访问的存储单元或 I/O 端口的地址,由 CPU 输出,以单向传输的方式进行。地址

线的数目决定了可以寻址的存储单元数量,例如,如果有 20 根地址线,那么可以寻址的存储单元数量为 2^{20} 。

(3) 控制总线(Control Bus)

控制总线负责传输协调各部件之间操作的控制信号。由于数据总线和地址总线是由所有挂在总线上的部件共享的,控制总线的作用是确保各部件能够在适当的时候获得总线的使用权。控制总线通常以单向传输的方式工作,例如,存储器的读/写命令或 I/O 设备的读/写命令都是由 CPU 发出的。然而,从整体上看,控制总线也可以是双向的,比如当某个设备准备好了,它会向 CPU 发送中断请求;或者当某个部件(如 DMA 接口)需要获得总线使用权时,它会向 CPU 发出总线请求。此外,控制总线还用于监控各部件的状态,比如查询设备是忙碌还是空闲,是否发生了错误等。

2. 时钟周期、机器周期及指令周期

(1) 时钟周期。计算机系统的有序操作依赖于时钟信号的精确节拍,这通常由内部时钟发生器电路提供。该电路产生的时钟信号的周期性间隔称为时钟周期,它是衡量处理器操作速度的基本单位。

(2) 机器周期。计算机执行最基本操作所需的时间长度,它由整数倍的时钟周期构成。例如,传统的 MCS-51 系列单片机中,一个机器周期由 12 个时钟周期组成,这反映了早期技术对操作速度的限制。然而,随着半导体技术的发展,现代微控制器(MCU)的设计已经能够实现机器周期与时钟周期的同步,即每个机器周期可能仅包含一个时钟周期。

(3) 指令周期。是指处理器执行一条指令所需的总时间,它由一个或多个机器周期组成。在具有复杂指令集的微处理器中,指令周期的长度会根据执行的具体指令而有所不同。

3. 指令系统

计算机的运作依赖于一系列指令的执行,这些指令共同构成了复杂的计算、决策和控制过程。CPU 作为计算机的大脑,其核心任务包括从存储器中提取指令和数据,进行必要的算术或逻辑处理,然后产生输出结果。而指令是 CPU 执行的基本操作的集合,每个指令都对应着计算机能够执行的一种特定操作。例如,从特定的存储单元中读取数据、将 CPU 内部寄存器的内容写入存储器的特定位置,或是执行算术和逻辑运算。这些操作以二进制形式编码,形成机器可以识别和执行的指令。

指令系统是计算机中所有可用指令的集合,它定义了 CPU 能够理解和执行的操作类型。不同的计算机架构拥有不同的指令系统,这取决于它们的设计和 functional 需求。在一个具体的指令系统中,每个指令都有一个唯一的二进制代码与之对应。为了让计算机精确地识别并实施指令中定义的任务,每种操作都需要有一个独特的指令代码来标识。计算机中所有这些指令的集合被称为指令集。

每条指令一般由两部分组成:操作码和操作数。操作码(Operation Code)指定了指令要执行的操作,通常用一个字节来表示。如果指令的种类非常多,可能就需要用两个字节来表示操作码(一个字节的操作码最多可以表示 256 种不同的操作)。操作数(Operand)则指明了参与操作的数据或者数据所在的存储单元地址。

计算机指令集拥有的指令种类、数目、指令代码对应的操作等部分,是由 CPU 的设计者定义的。因此,不同类型的 CPU 通常会有不同的指令集,不同系列的 CPU 拥有不同的指令

集,除非它们是相互兼容的。根据指令集的特点,计算机的指令集可以被划分为两大类:复杂指令集计算机(Complex Instruction Set Computer,CISC)和精简指令集计算机(Reduced Instruction Set Computer,RISC)。

一般而言,计算机执行一条指令的过程,可为以下几步:

(1)取操作码:CPU 从程序计数器 PC 指向的存储单元中读取指令的第一字节,即操作码。

(2)译码:指令译码器对取得的操作码进行解析,确定指令的含义,并指示控制器生成相应的控制信号。

(3)取操作数:根据操作码指示,CPU 读取指令的剩余字节,这些字节通常包含操作数的信息或操作数的地址。

(4)执行:CPU 执行指令规定的操作,这可能包括算术运算、逻辑运算、数据传输等。

在整个指令取出和执行的过程中,程序计数器 PC 是跟踪即将执行的指令的位置。每当 CPU 从存储器中取出一个指令或操作数后,PC 的值就会自动增加,指向下一个待执行指令的地址。这个过程会不断重复,直到程序执行完毕。

简而言之,CPU 通过不断循环“取操作码→译码→取操作数→执行”的步骤来逐条执行程序中的指令,而 PC 确保 CPU 知道下一步应该执行哪一条指令。

1.1.2 计算机系统的内部结构

1. CPU 的内部结构

计算机系统的心脏是中央处理器(Central Processor Unit,CPU),它负责执行程序中的指令和处理数据。CPU 的内部结构相当复杂,但核心部件主要包括运算器和控制器。一个 8 位通用微处理器内部基本结构可用图 1-2 描述,它由算术逻辑运算单元 ALU(Arithmetic Logic Unit)、累加器 A(8 位)、寄存器 B(8 位)、程序状态字寄存器 PSW(8 位)、程序计数器 PC(也称为指令指针,即 IP,16 位)、地址寄存器 AR(16 位)、数据寄存器 DR(8 位)、指令寄存器 IR(8 位)、指令译码器 ID、控制器等部件组成。

(1)程序计数器(Program Counter,PC)是中央处理器(Central Processing Unit,CPU)内部的一个特殊寄存器,它的主要作用是追踪即将要执行的指令所在的内存地址。PC 寄存器的大小通常与 CPU 能够访问的地址线的数目一致。例如 8 位微机 CPU 一般具有 16 根地址线(A15~A0),PC 的长度也是 16 位。当系统启动或复位时,PC 寄存器会被赋予一个初始值,这个值指向了程序存储器中的第一条指令的起始位置,以 MCS-51 系列单片机为例,复位后 PC 寄存器的值被设定为 0x0000,意味着 CPU 将从存储器的这个地址开始读取并执行指令。在程序的执行过程中,每当 CPU 从存储器读取一条指令后,PC 寄存器会自动增加 1(或者说增加指令所占的字节数),以便指向下一条指令的地址。这种自动递增的特性,使得程序能够顺序地执行存储在内存中的指令序列。如果程序中包含跳转指令,PC 寄存器的值会根据跳转逻辑更新,以指向新的执行位置。

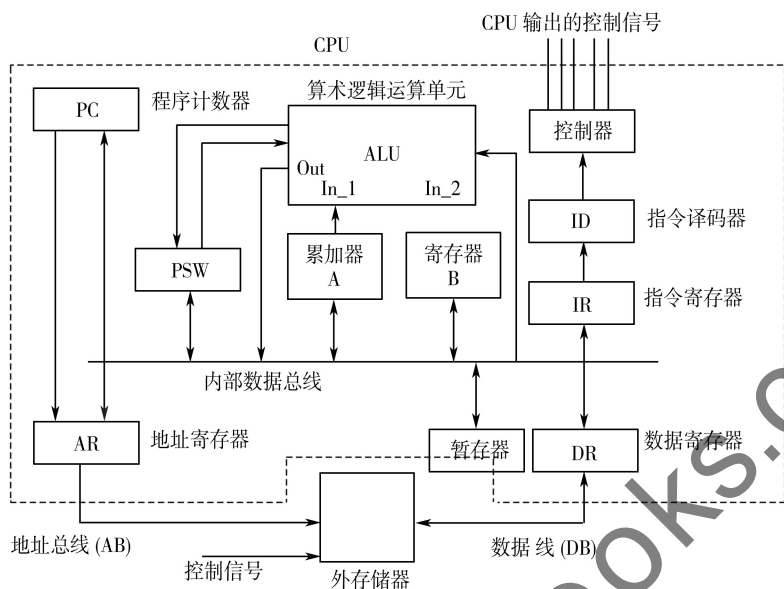


图 1-2 CPU 内部结构简图

(2) 地址寄存器 AR(Address Register) 用于存储即将访问的外部存储器单元的地址。它的作用是帮助 CPU 定位下一个要读取或写入的数据的位置, 指令码所在存储单元的地址编码一般由程序计数器 PC 产生, 而指令中操作数所在存储单元的地址码由指令的操作数给定。地址寄存器通过地址总线(AB)与系统的外部存储器进行通信。

(3) 指令寄存器 IR(Instruction Register) 用于临时存储从存储器中读取的指令代码的第一个字节, 也就是操作码。这个操作码随后会被送入指令译码器(ID), 译码器会解析操作码并生成相应的控制信号, 这些信号指导 CPU 执行指令所规定的操作。

(4) 数据寄存器 DR(Data Register) 用于存储将要写入外部存储器或输入/输出(I/O)端口的数据。数据寄存器具有锁存功能, 这意味着它可以保持(锁存)所接收的数据, 直到该数据被写入到目的地为止。数据寄存器与外部数据总线(DB)相连, 这使得数据可以在 CPU 和系统其他组件之间传输。

(5) 算术逻辑运算单元 ALU 主要用于基本算术和逻辑运算。ALU 负责的算术运算包括加法、减法、乘法和除法, 而逻辑运算则包括 AND(与)、OR(或)、NOT(非)和 XOR(异或)。由于 ALU 本身不包含寄存器, 所以需要使用 CPU 中的加器 A 进行运算时, 完成操作数的暂存和运算结果的存储。

(6) 程序状态字寄存器(Program Status Word, PSW) 用于记录运算过程中的一些状态信息, 比如是否有溢出或进位发生。

总之, CPU 作为计算机系统的核心部件, 需要负责指令的控制, 指令的操作, 时间的控制, 数据的加工和中断的处理等相关功能。

2. 存储器

存储器是计算机系统中必不可少的存储设备, 主要负责存储程序指令和数据。尽管 CPU 内部的寄存器也用于存储信息, 但这些寄存器数量有限且速度较快, 主要用于暂存参与运算的操作数和中间结果。存储器通常位于 CPU 外部, 但单片机 CPU 通常内部会集成

一定容量的存储器,存储器芯片包含大量的存储单元,其容量范围从几千字节到数千兆字节,能够存储庞大的信息量,然而,与 CPU 内部寄存器相比,存储器的存取速度较慢。目前,存储器的存取速度已成为影响计算机整体运行速度的一个关键因素。

存储器主要分为两大类:只读存储器(Read Only Memory,ROM)和随机读写存储器(Random Access Memory,RAM)。ROM 中的数据一旦写入就无法更改,适合存储系统监控程序。而 RAM 允许随机读写,但其数据在断电后会丢失,因此适合存储会频繁修改的数据。

根据存储器存储单元结构和信息保存方式,RAM 可以进一步分为静态 RAM(SRAM)和动态 RAM(DRAM)。SRAM 由双极型晶体管或 MOS 管构成,存取速度快,不需要刷新,但由于每个存储单元需要较多的晶体管,因此集成度较低,价格也较高。DRAM 则依赖 MOS 管栅极与衬底之间的寄生电容来存储信息,多为单管结构,集成度高,但寄生电容容量小,漏电大,信息保存时间短,仅为毫秒级,需要定期刷新以避免数据丢失,这使得其电路设计相对复杂。

在单片机系统中,由于对存储容量的需求通常不大,且为了简化外围电路,几乎很少使用 DRAM。常用的存储器类型包括 PROM(可编程 ROM)、EPROM(可擦写可编程 ROM,现多被 OTPROM 和 Flash ROM 取代)、OTPROM(一次性编程 ROM,无擦除窗口)、EEPROM(电可擦写可编程 ROM,具有较薄的绝缘栅,可通过高电压擦除)和 Flash ROM(电可擦写 ROM,写入速度快,也称为闪存)。尽管这些存储器的工作原理不同,但内部结构基本相同。

EEPROM 和 Flash ROM 的主要区别在于擦写次数和方式。EEPROM 可以承受超过 30 万次的擦写,远高于 Flash ROM 的 1 万次;EEPROM 支持单字节的擦除和写入,而 Flash ROM 通常需要按块方式进行擦除和写入。因此,在微控制器系统中,Flash ROM 通常用于存储系统控制程序,而 EEPROM 则用作存储非易失性数据的存储器。

(1) 内部结构

EPROM、EEPROM、Flash ROM、SRAM、FRAM 等存储器内部结构可以用图 1-3 描述,由地址译码器、存储单元、读写控制电路等部分组成。

在计算机的寄存器或存储器中,一个存储单元的功能可以类比为—组触发器。每个触发器具备两种稳定的状态,分别代表二进制数字 0 和 1,因此可以存储一个比特的信息。存储单元的“字长”是指该单元中包含的触发器数量,它决定了存储单元能够存储的信息量。对于可以并行访问的存储器芯片来说,存储单元中触发器的数量与存储器芯片的数据线条数一致。

例如,如果一个存储器芯片的数据总线宽度为 8 位,那么它的每个存储单元就包含 8 个触发器,这样就可以存储 8 位二进制数,也就是一个字节的数据。这样的设计允许存储器在单一操作中同时存取多个比特,从而提高数据的处理速度和效率。

存储器芯片中存储单元的数量与地址线数目有关。例如,一个拥有 15 条地址线的存储器芯片可以寻址 32,768 个存储单元,也就是 32KB(215B)的存储容量。每个存储单元都会被赋予一个唯一的地址,以便 CPU 或其他部件能够访问和控制这些单元。这些地址通常是连续的,从最低地址开始一直到最高地址。例如,一个 32KB 的存储器芯片可能拥有从

0000H 到 7FFFH 的地址范围。

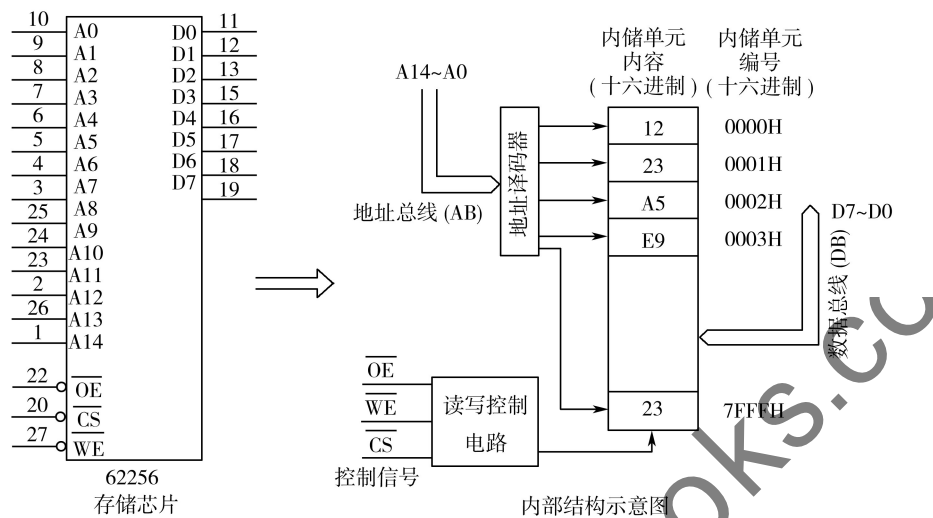


图 1-3 存储器芯片及内部结构

存储单元的地址编码是独立于存储单元内容的,地址编码仅用于标识存储单元的位置,存储长度由存储器芯片所包含的存储单元的个数决定。存储单元的内容是指存储在该单元中的二进制信息,它们可以是指令、数据或其他信息。在设计存储器系统时,需要考虑存储单元的大小、数量和寻址方式,以满足系统的性能和容量需求。

(2) 存储器工作状态

存储器芯片的工作状态由存储器控制信号的电平状态决定,如表 1-1 所示。

表 1-1 存储器工作状态

工作模式	控制信号			输出
	片选信号 \overline{CS}	输出允许信号 \overline{OE}	写允许信号 \overline{WE}	
读	L	L	H	数据输出
输出禁止	L	H	H	高阻态
待用(功率下降)	H	X	X	高阻态
写入	L	H	L	数据输入

(3) 存储器读操作

下面是个 CPU 如何从存储器中读取特定地址单元的数据的例子,阐述了 CPU 对地址编号为 0x0000 的存储单元进行数据读取的过程,具体步骤如下:

- ① CPU 首先通过地址寄存器(AR)确定要访问的存储单元的地址,本例中为 0x0000。
- ② CPU 将这个地址通过地址总线(A15 至 A0)发送给存储器。CPU 地址总线与存储器的地址总线相连,确保地址信息正确传递。
- ③ 存储器芯片内的地址译码器接收到 0x0000H 单元地址信号后,对这些信号进行解码和识别,并选中对应的存储单元。
- ④ CPU 发出一个读控制信号(RD),通常这个信号会连接到存储器的输出使能端(OE)。

被选中的存储单元 0x0000 中的数据(假设为 0x12)随后被放到数据总线(D7 至 D0)上,并通过这条总线传输到 CPU 内部的数据寄存器(DR)中,之后,这个数据会被传输到 CPU 内部的某个特定寄存器或暂存器中。这样便完成了读操作的过程。

(4) 存储器写操作

关于写入操作,就是如何将数据写入存储器中的特定单元。以图 1-4 为例,我们要将数据 55H 写入地址为 0003H 的存储单元,具体步骤如下:

①CPU 的地址寄存器(AR)先确定要写入的存储单元地址,本例中为 0003H。这个地址随后通过地址总线(A15 至 A0)发送到存储器芯片的地址线(A14 至 A0)上。

②存储器芯片内的地址译码器接收到地址信号后,会进行解码以识别并选中 0003H 存储单元。

③在写入过程中,待写入的数据 55H 首先存放在 CPU 内部的数据寄存器(DR)中。当 CPU 发出写控制信号(\overline{WR}),并且该信号有效(与存储器的写允许信号 \overline{WE} 相连),DR 中的数据 55H 就会通过数据总线(D7 至 D0)传输到之前选中的存储单元 0003H 中。这样,存储单元 0003H 的内容就会被更新为 55H,从而完成写操作。

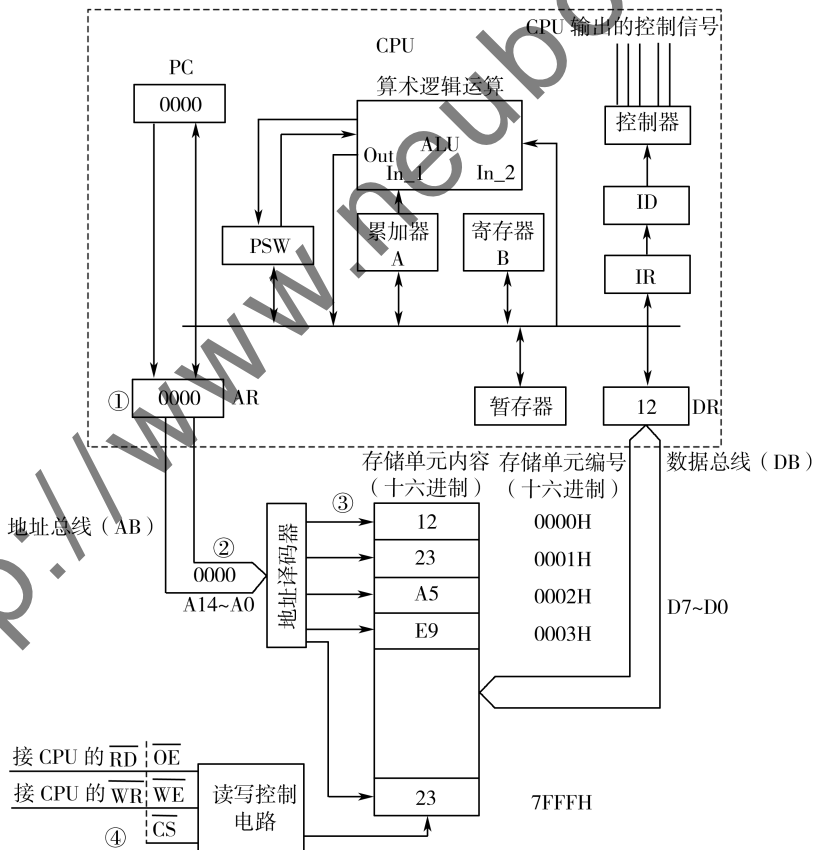


图 1-4 CPU 读取存储器操作过程示意图

值得注意的是,一旦执行了写操作,存储单元中原有的信息就会被新写入的数据覆盖,原有的数据将不再存在。这一过程确保了数据能够被准确地存储在指定的位置。

3. I/O 设备

中央处理器(CPU)和主存储器(主存)共同构成了计算机的主机部分。除了主机之外,其余的硬件设备通常被称为输入/输出设备(I/O 设备)、外部设备或者简称为外设。没有输入输出设备的计算机系统就像没有软件的计算机系统一样,失去了其存在的价值和功能。

随着计算机技术的持续进步,I/O 设备在计算机系统中变得越来越重要,它们在整个系统成本中所占的比例也在不断增加。在早期的计算机系统中,主机结构相对简单,处理速度较慢,应用范围有限,所配备的 I/O 设备种类较少,数量也不多,I/O 设备的成本仅占整个系统成本的很小一部分,通常只有几个百分点。然而,现代计算机系统的 I/O 设备已经向多样化和智能化的方向发展,种类繁多,性能优越,其成本已经占据了系统总成本的大约 80%左右。这表明,I/O 设备已成为计算机系统中不可或缺的重要组成部分,并且对整个系统的性能和价值有着决定性的影响。

I/O 设备可以根据其功能和用途被分为三大类:

(1) 人机交互设备

这类设备是用户与计算机系统之间进行信息交流的桥梁。它们能够将人的感官可识别的信息(如视觉、听觉等)转换为计算机能够处理的数字信号,或者反过来,将计算机的处理结果转换为人类可识别的信息。包括输入设备,比如键盘、鼠标、手写板、扫描仪、摄像机、语音识别器等,它们允许用户将信息输入到计算机中;还有输出设备,比如打印机、显示器、绘图仪、语音合成器等,它们将计算机的处理结果以人类可理解的形式展示出来。

(2) 计算机信息存储设备

这些设备用于存储大量的系统软件、应用程序和数据,因为这些信息通常需要长期保存。存储设备通常作为计算机系统的辅助存储器,包括磁盘(如硬盘驱动器和固态硬盘),光盘(如 CD、DVD、蓝光光盘)和磁带(用于数据备份和归档)。

(3) 机-机通信设备

这类设备使得计算机能够与其他计算机或系统之间进行数据交换和通信。它们在构建网络和实现实时控制等应用中至关重要。包括调制解调器(Modem),通过电话线实现计算机之间的通信;数据通信接口(如 DIA、AID 转换设备),用于实时工业控制,实现数据的采集和控制信号的转换;网络设备,实现计算机与计算机以及其他系统的远距离信息交换。

1.1.3 计算机的主要技术指标

评估一台计算机的性能是否优越,需要综合考虑多个技术和性能指标。这些指标不仅包括硬件方面的参数,还涉及软件的功能和效率。

1. 机器字长

机器字长定义了 CPU 在一次操作中能够处理的数据量,它通常与 CPU 内部寄存器的大小相对应。字长越长,意味着 CPU 能够一次性处理更多的位数,这使得计算机能够表示更大的数值范围,并提高数值计算的精度。例如,一个 32 位的 CPU 可以处理比 16 位 CPU 更大范围的整数和更高精度的小数。

然而,增加字长也会带来一些影响。首先,它会影响计算机的运算速度。如果 CPU 的字长较短,而需要处理的数据位数较多,那么可能需要通过多次运算来完成,这会降低处理

速度。其次,较长的字长会增加硬件成本,因为需要更多的硬件资源来支持更宽的数据路径和存储空间。例如,加法器(或算术逻辑单元 ALU)、数据总线和存储器的位数都需要与 CPU 的字长相匹配。

因此,在确定机器字长时,需要综合考虑多个因素,包括但不限于数据处理的精度需求、数值表示的范围、运算速度的要求以及硬件成本。设计者需要在这些因素之间做出平衡,以满足特定应用场景下的性能和成本要求。例如,服务器和高性能计算应用可能需要较长的字长来处理大量数据和进行复杂计算,而对于成本敏感的嵌入式系统或移动设备,可能会选择较短的字长以降低成本。

2. 存储容量

存储器容量指的是计算机系统中用于存储数据和程序的总空间,它由两部分组成:主存储器(主存)和辅助存储器(辅存)的容量。

主存容量:主存是 CPU 直接访问的存储空间,其容量指的是主存中可以存储的二进制代码的总位数。主存容量的计算公式为:

$$\text{存储容量} = \text{存储单元个数} \times \text{存储字长}$$

其中,存储单元个数表示主存中存储单元的总数,存储字长表示每个存储单元可以存储的位数。现代计算机通常以字节为单位来描述存储容量,由于一个字节等于 8 位二进制代码,因此主存容量可以用字节数来表示。

辅存容量:辅存是用于长期存储数据和程序的存储设备,它作为主存的补充,提供更大的存储空间。辅存的容量通常也以字节为单位来表示,以便于描述和理解。例如,硬盘、固态硬盘、光盘、USB 驱动器等都是常见的辅存设备。辅存的容量通常远大于主存,可以达到数十 GB 甚至 TB 级别。

在描述辅存容量时,常用的单位转换为:

$$1 \text{ KB (千字节)} = 1,024 \text{ 字节}$$

$$1 \text{ MB (兆字节)} = 1,024 \text{ KB}$$

$$1 \text{ GB (吉字节)} = 1,024 \text{ MB}$$

$$1 \text{ TB (太字节)} = 1,024 \text{ GB}$$

3. 运算速度

计算机的运算速度受多种因素影响,包括但不限于主频(CPU 的时钟频率,即 CPU 每秒可以执行的周期数),操作类型(如算术运算、逻辑运算等,可能需要不同数量的周期来完成),主存速度(主存的存取速度影响指令和数据的获取速度)。

早期衡量运算速度的方法是通过一次加法或乘法操作所需的时间,这种方法存在局限性。随后,引入了吉普森(Gibson)法,该方法更为综合,考虑了每条指令的执行时间和它们在整个程序中所占的比例,计算公式为:

$$T_M = \sum_{i=1}^n f_i t_i$$

其中:

— T_M 是机器的运行速度。

— f_i 是第 i 种指令占全部操作的百分比。

— t_i 是第 i 种指令的平均执行时间。

当前,计算机的运算速度通常用单位时间内执行的指令数量来衡量,使用 MIPS(百万条指令每秒)作为计量单位。例如,如果一台计算机每秒能执行 200 万条指令,则其运算速度记为 2 MIPS。此外,还可以使用以下指标来衡量运算速度:

- CPI(每条指令的时钟周期数):执行一条指令所需的平均时钟周期数,也可以表示为机器主频的倒数。

- FLOPS(每秒浮点运算次数):计算机每秒可以执行的浮点运算次数,是衡量计算机处理浮点运算能力的指标。

这些指标提供了不同角度的衡量方法,允许用户根据特定的需求和应用场景评估计算机的性能。

1.2 嵌入式系统概述

在当今这个数字信息和网络技术迅猛发展时代,嵌入式系统已经深入到科学研究、工程设计、军事技术以及我们日常生活的各个角落。这种系统由软件和硬件共同构成,其核心部件是嵌入式处理器和嵌入式操作系统。

1.2.1 嵌入式系统的基本概念

在计算机的早期发展阶段,人们习惯于根据计算机的架构、计算速度、系统规模和应用范围等因素,将其划分为大型机、中型机、小型机和个人电脑。这种分类方式一直沿用到了 20 世纪 90 年代中期。但是,随着半导体技术和计算机技术的迅猛发展,传统的分类方法已经不能满足当前的实际情况。例如,现代的个人电脑(PC),尽管起源于小型机,但其在处理能力、总线设计、内存寻址等方面与最初的小型机定义已经大相径庭。事实上,它们的整体性能已经超越了当年所定义的中型和小型机。另外,随着计算机技术在不同领域的深入融合,逐渐以应用为核心的方式将计算机划分为:嵌入式计算机和通用计算机两大类。

那么,什么是嵌入式系统呢?嵌入式系统是一种以特定应用为核心,以计算机技术为支撑,集成了硬件和软件的系统,旨在满足特定的功能、可靠性、成本、体积和能耗等要求。这种系统通常嵌入在机械或电子设备中。其硬件部分包括嵌入式处理器、控制单元、数字信号处理器(DSP)、存储设备和外围设备接口等。软件部分则包括启动程序、嵌入式操作系统和应用程序等。嵌入式操作系统负责协调应用程序与硬件之间的交互,而应用程序则负责控制整个系统的运行。有时,嵌入式系统还会包含一些机械组件,如机电一体化设备、微电子机械系统(MEMS)、光学系统等,这些组件是为了实现特定功能而设计的。因此,嵌入式系统有时也被称为嵌入式设备。

1.2.2 嵌入式系统的发展及特点

嵌入式系统的发展可以追溯到 20 世纪 60 年代,当时它们被用于控制电话交换机,被称为“存储式过程控制系统”(Stored Program Control System)。现代意义上的嵌入式系统大

约在 1970 年出现,至今已经有超过 40 年的发展历程。嵌入式技术的演进大致可以分为以下几个阶段:

第一阶段:以单个芯片为核心的系统,这些系统是可编程的控制器,与监测、伺服、指示设备等配合工作。它们通常应用于专业性强的控制领域,没有操作系统支持,直接通过汇编语言进行控制。这个阶段的系统结构和功能相对单一,处理效率不高,存储容量有限,用户界面几乎不存在。在中国工业领域,这种系统曾经非常普遍。

第二阶段:基于嵌入式 CPU 和简单操作系统的系统。这一阶段的特点是 CPU 种类多样,但通用性不强;系统开销小,效率高;操作系统具有一定的兼容性和可扩展性;应用软件专业化,用户界面尚不够友好。

第三阶段:以嵌入式操作系统为核心的系统。这一阶段的系统能够运行在各种不同类型的微处理器上,具有良好的兼容性;操作系统内核小、效率高,具有高度的模块化和可扩展性;系统具备文件和目录管理、多任务处理、网络支持、图形用户界面等功能;提供大量应用程序接口 API,简化了应用程序的开发;嵌入式应用软件变得丰富多样。

第四阶段:以互联网为核心的嵌入式系统。这是一个快速发展的阶段,目前大多数嵌入式系统还未与互联网连接,但随着互联网技术的发展,以及与信息家电、工业控制技术的紧密结合,嵌入式设备与互联网的结合预示着嵌入式系统的未来方向。

在数字化和网络技术迅猛发展的后 PC 时代,嵌入式系统已经深入到科学研究、工程设计、军事技术、工业、商业、文化艺术乃至日常生活的方方面面。随着嵌入式产品的不断开发和普及,嵌入式技术与人们的日常生活越来越紧密地联系在一起。而且,嵌入式系统是一种为特定应用量身定制的计算机应用系统,其核心特征在于其专用性,这类系统的功能和非功能指标,如外形和体积,都是经过特别设计的,并且具有很低的冗余度。由于嵌入式系统应用广泛且需求多样,导致不同嵌入式系统的软硬件复杂度存在显著差异,这使得该行业难以被单一实体所垄断。

因此,与通用计算机相比,嵌入式系统具有以下显著特点:

- 嵌入式系统是专门设计的计算机系统,通常包含针对特定应用的嵌入式中央处理单元(CPU)。
- 它们是将先进的计算机技术、半导体技术和电子技术与具体行业应用相结合的成果,因此可以认为嵌入式系统不属于单一学科,而是跨越了多个学科领域。
- 嵌入式系统的硬件和软件设计必须实现低功耗、小体积、高集成度和低成本的目标。
- 嵌入式系统与具体应用紧密结合,运行环境差异显著,并且具有较长的使用寿命周期。
- 为了提升执行速度和系统稳定性,嵌入式软件系统通常固化在非易失性存储器中。
- 嵌入式系统本身不具备自主开发能力,因此需要交叉编译,这意味着必须有一套特定的开发工具和环境才能进行开发。

1.2.3 嵌入式系统的分类

嵌入式系统与嵌入式处理器是两个不同的概念。嵌入式系统指的是一个完整的嵌入式计算机系统,它不仅包含嵌入式处理器,还集成了其他必要的硬件和软件组件,以形成一个

能够满足特定嵌入式应用需求的系统。只有当嵌入式处理器与其他组件结合,构成一个计算机系统,并被应用于嵌入式环境时,这个整体才能被称为嵌入式系统。

嵌入式系统与其应用的对象系统紧密相关,其主要技术发展方向是不断扩展和完善,以满足对象系统的要求。这包括增加各种外围电路,如模数转换器(ADC)、数模转换器(DAC)、脉冲宽度调制(PWM)、实时时钟、电源监控、程序运行监控电路等,以形成一个满足特定应用需求的系统。因此,作为一个专用的计算机系统,嵌入式系统需要不断地向更广泛的计算机应用系统发展。

根据不同的分类标准,嵌入式系统可以按照以下方式进行分类:

(1)按硬件范畴分类

- ①芯片级嵌入式系统:处理器芯片内部集成了程序或算法。
- ②模块级嵌入式系统:系统包含一个或多个核心功能模块。
- ③系统级嵌入式系统:包含完整的系统组件,并集成了嵌入式软件的所有内容。

(2)按软件范畴分类,特别是根据系统的实时性

①实时系统:能够及时响应外部随机事件,并以足够快的速度完成事件处理的计算机应用系统。实时系统的一个重要特点是,如果系统逻辑或时序出现问题,可能会导致严重的后果。实时系统进一步细分为:软实时系统(对时间要求相对宽松,偶尔的延迟不会导致灾难性后果);硬实时系统(对时间要求非常严格,任何延迟都可能导致系统故障)。

②非实时系统:用于对外部响应时间要求不严格的应用,例如个人助理设备(PAD)等。

1.2.4 嵌入式系统的组成

一个嵌入式系统由4个部分组成:嵌入式处理器、嵌入式外围设备、嵌入式操作系统和嵌入式应用软件。嵌入式系统的组成结构如图1-5所示。

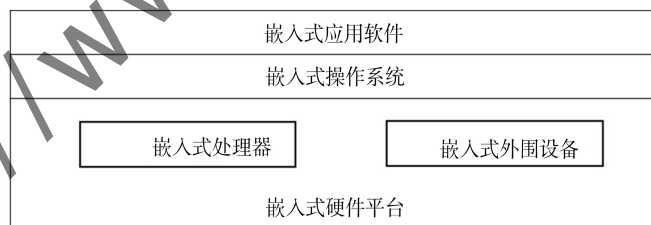


图 1-5 嵌入式系统的组成结构

硬件构成了软件运行的物理平台,而嵌入式系统的硬件架构则以嵌入式处理器为核心,辅以存储器、输入/输出(I/O)单元、通信接口以及必需的外围设备等辅助接口。如图1-6所示,嵌入式系统的硬件配置在实际应用中可以非常简洁,除了必不可少的微处理器和基础外围电路外,其他电路可以根据需求和成本效益进行选择性地削减(例如,虚线部分所示的模块)和个性化定制。此外,嵌入式系统通常还包含用户界面,通过键盘、液晶显示屏、触摸屏等设备实现与用户的交互。

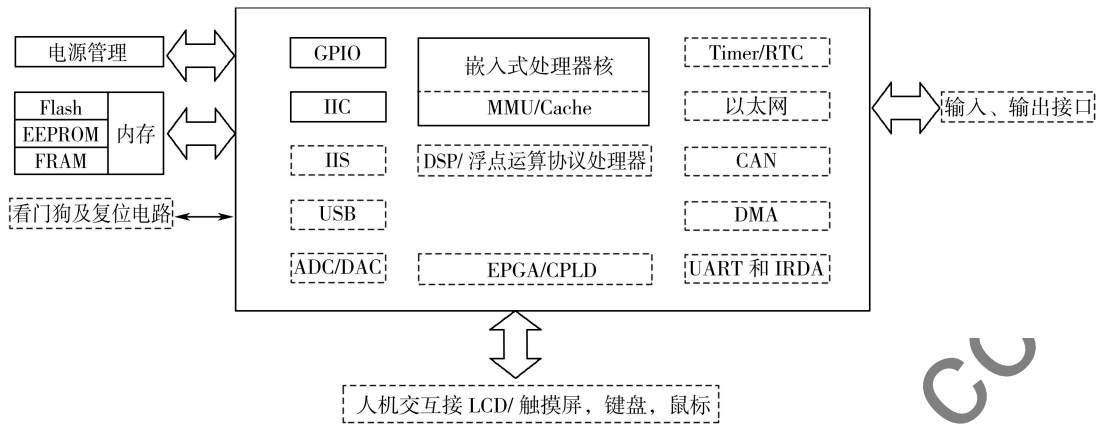


图 1-6 嵌入式系统硬件组成

嵌入式系统的存储设备根据系统规模和需求而异。许多微控制器(MCU)系统通常使用内置的 Flash 和 SRAM 存储器,代码被固化在 Flash 中,可以直接执行,无须额外的存储扩展。而微处理器(MPU)系统则需要扩展 SDRAM 作为主内存,并使用 NOR Flash 作为外存储器。程序通常存储在外存储器中,在需要执行时由操作系统(OS)调入内存。

嵌入式处理器一般可分为四类:嵌入式微控制器(MicroController Unit,MCU)、嵌入式微处理器(MicroProcessor Unit,MPU)、数字信号处理器(Digital Signal Processor,DSP)和片上系统(System on Chip,SoC)。

除了嵌入式处理器和存储器,嵌入式系统的硬件部分还包括各种外围设备接口。这些丰富的接口使得嵌入式系统能够实现多样化的应用。例如,现代的 ARM 微控制器拥有非常丰富的外设接口,如 I2C、SPI、UART 和 USB 等,这些接口已成为标准配置。在系统设计时,只需将微控制器与相应接口的外部设备进行简单的物理连接,即可实现外设扩展。

嵌入式系统的软件通常由嵌入式操作系统和应用软件组成。具体来说,嵌入式系统软件可以分为启动代码(Boot Loader)、板级支持包(Board Support Package,BSP)、操作系统内核与驱动(Kernel & Driver)、文件系统与应用程序(File System & Application)等部分。

Boot Loader 是嵌入式系统的启动代码,主要用于初始化处理器,并将内核启动参数传递给嵌入式操作系统内核,以便内核能够根据参数要求启动。操作系统内核主要有四个任务:进程管理、进程间通信与同步、内存管理和 I/O 资源管理。驱动程序也是内核的一部分,它为上层应用程序提供了通过处理器外设接口控制器与外部设备通信的媒介。文件系统允许嵌入式软件工程师灵活方便地管理和使用系统资源,包括其他系统软件/中间件,如图形用户界面(GUI)、网络协议栈等。应用程序则是针对特定需求开发的,可能是嵌入式软件工程师完全自主开发的。

总的来说,嵌入式系统的硬件部分是整个系统的基石,而软件部分则是在此基础上构建的不同功能模块。对于任何一个具有明确需求的嵌入式系统来说,硬件和软件都是不可或缺的。

1.3 STM8 单片机简介

在2009年,STMicroelectronics(意法半导体公司)推出了基于先进8位架构的STM8系列微控制器。这个系列针对不同的应用场景,主要包含三个子系列:STM8L、STM8S和STM8A。STM8L系列专为超低功耗设计,适合于对电池寿命有高要求的应用,如便携式设备、医疗设备、工业设备、电子计量设备、感应器或安全设备等。STM8S系列则主要面向工业制造和消费电子产品,适用于需要较高性能和灵活的I/O支持的场合。STM8A系列则是专为汽车应用设计的,能够满足汽车行业对半导体产品的苛刻要求,如耐高温、抗干扰等特性。

1.3.1 STM8 单片机的特点

STM8系列微控制器采用了哈佛架构,专为特定的应用领域量身定制,包括汽车、工业、低电压以及电池驱动的设备,同时还满足了特定应用的标准化需求。这一平台以其卓越的性能、稳定性、缩短的设计时间和降低的总体系统成本而著称。在8位微控制器市场中,STM8平台的设计理念集中在如何有效减少解决方案的总体成本,同时提供更强大的性能。因此,STM8平台在软件和硬件成本、处理速度、性能以及成本效益之间实现了优化的平衡,使其成为一款具有高成本效益的微控制器。下面我们将深入探讨STM8平台的多项优势。

①STM8系列单片机基于哈佛架构设计,配备了16位索引寄存器和堆栈指针,支持高达16MB的线性地址空间,并采用先进的寻址模式,这些特点共同优化了C语言程序的执行速度和代码紧凑度。其内核处理速度平均每条指令仅需1.6个时钟周期,在24MHz时钟频率下,性能可达到20MIPS。STM8采用了3级流水线技术,进一步提升了处理效率。

②STM8单片机内部集成了大容量、高品质的Flash程序存储器,最大容量可达128KB,支持ISP和IAP编程,便于产品在调试、开发、生产和更新过程中的可擦写操作。STM8平台提供的嵌入式EEPROM存储器具有高耐擦写和数据保持能力,与外部存储器相媲美,从而简化了基于闪存的复杂仿真方法。此外,片内大容量的SRAM不仅满足了一般应用需求,还有效支持了高级语言开发的系统程序,并通过外部总线扩展外部RAM。

③STM8单片机的I/O线可灵活配置为浮动输入、带内部上拉输入、推挽输出和开漏输出模式,并可设置输出摆率控制以降低EMC噪声。其输入/输出引脚设计能够承受强烈的外部干扰,无需外部保护元件即可实现强大的抗干扰性能。

④STM8单片机内部还具备灵活的时钟控制系统,包括4个主时钟源和带有时钟监控的时钟安全系统(CSS)。它提供多种独立的时钟分频器,为Timer、UART、SPI、I2C、ADC、AWU、CAN以及看门狗等外设提供定制的时钟需求。特别是定时器TIM1,具备高达16位的预分频器,支持通过软件设置分频系统,提供多样化的定时需求,支持6路PWM输出,并

支持边缘和中心对齐模式,以及互补输出和死区时间控制。

⑤ STM8 单片机增强型的全双工高速同步/异步串口,具备硬件生成校验码、硬件检测和校验功能,支持 LIN 模式、IrDA、SIR 编码和解码,以及 SmartCard 模拟功能。

⑥ STM8 单片机还具备自动上电复位电路、掉电复位电路、独立看门狗电路、窗口看门狗电路和多个复位源(包括 NRST 引脚产生的外部复位、上电复位、掉电复位、看门狗复位、软件复位、SWIM 复位、非法操作码复位和 EMS 复位)。这些特性增强了嵌入式系统的可靠性,并且可设置的启动延时运行程序进一步保障了系统的稳定运行。

⑦ STM8 单片机还支持多种低功耗模式,包括等待、活跃停机和停机模式,外设时钟可以单独关闭,且能在 1.65~5.5V 的宽电压范围内运行,这为电源管理提供了极大的灵活性。

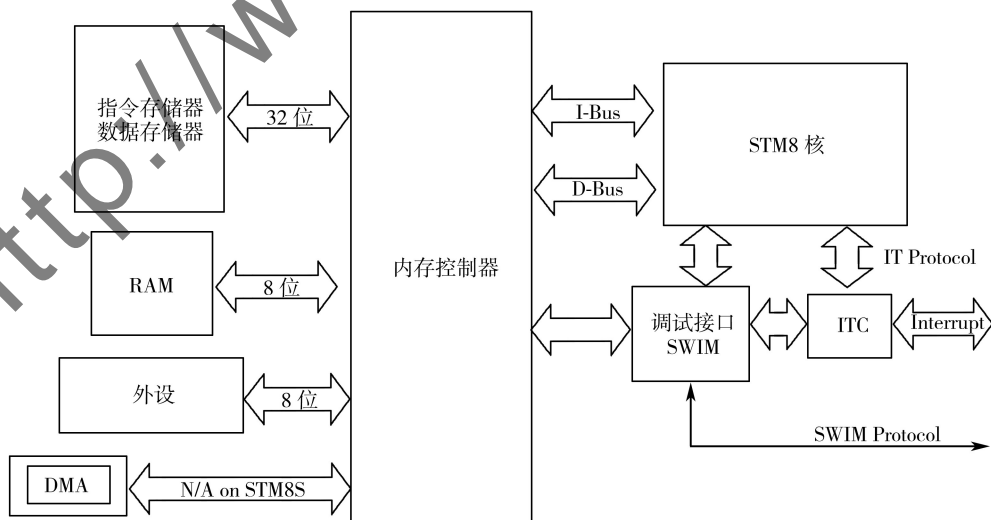
⑧ 此外,STM8 单片机配备了单线接口 SWIM 和调试模块 DM,便于进行在线编程和非侵入式调试。

综上所述,STM8 单片机具有高性能、灵活性和可靠性的特点,在 8 位单片机市场中具有较高的性价比,适用于工业、农业和消费电子等多个领域。

1.3.2 STM8 单片机结构

STM8 系列微控制器的基本组成结构,如图 1-7 所示,涵盖了多个关键组件:STM8 核心处理器、中断控制单元 ITC、内存管理单元、SWIM 调试接口、外设接口、随机存取存储器 RAM、指令存储器、数据存储器以及 OMA 模块。值得注意的是,在 STM8S 系列中并不包含 OMA 模块。

STM8 核心处理器是一个 8 位的 CPU,配备了一个 32 位的存储器接口和三级流水线架构,这使得它在 24MHz 的操作频率下能够达到最高 20 MIPS 的处理性能。ITC 作为中断控制单元,负责管理中断,其中所有的 I/O 引脚都能够触发外部中断,每个端口都配备了独立的中断向量和状态标志,并且支持最多 4 层的软件可编程中断优先级。



1.3.3 STM8 系列单片机分类

1. STM8S 单片机系统产品

STM8S 平台代表了 8 位微控制器领域的新发展,其 CPU 性能高达 20MIPS,并且支持 2.95~5.5V 的电压范围,这使得现有的 8 位系统能够更容易地过渡到使用更低电压的电源。该平台采用的 130nm 非易失性存储器技术是当前 8 位微控制器中最先进的技术之一,它提供了真正的 EEPROM 数据写入操作,且具有高达 30 万次的擦写寿命。在家用电器、暖通空调系统、工业自动化、电动工具、个人护理设备以及电源控制管理系统等众多产品和设备中,STM8S 平台所集成的丰富外设能够支持精确的控制和监视功能。这些功能包括:10 位模数转换器,最多可提供 16 个通道,转换时间小于 3 微秒;先进的 16 位控制定时器,适用于电机控制、捕获/比较以及 PWM 功能。其他外设包括 CAN 2.0B 接口、两个 USART 接口、一个 I2C 端口和一个 SPI 端口。

此外,STM8S 平台的外定义与 STM32 系列 32 位微控制器相同,这种共通性提高了不同产品间的兼容性,并为设计提供了更大的灵活性。应用代码可以移植到 STM32 平台,以获得更高的性能。STM8S 的组件和封装在引脚上完全兼容,这为开发人员提供了更大的自由度来优化引脚数量和外设性能。引脚的兼容性还有助于平台化设计决策,可以节省产品上市时间并简化产品升级过程。

STM8S 主要特点:

- 卓越的处理能力:STM8S 搭载的内核性能强劲,处理速度高达 20MIPS,确保了快速的数据处理。
- 强大的抗干扰性能:该系列微控制器具备出色的抗干扰能力,保证了在复杂环境下的稳定运行和高可靠性。
- 先进的制造技术:采用 130 纳米工艺制造,STM8S 在性能和成本效益之间实现了卓越的平衡。
- 广泛的产品线:STM8S 提供从 4K 到 128K 的程序存储空间,以及从 20 脚到 80 脚的多重芯片封装选择,满足不同规模的应用需求。
- 经济的系统成本:内建 EEPROM 和高精度 RC 振荡器,有效降低了系统整体的物料和开发成本。
- 便捷的开发体验:得益于本地化的工具支持,STM8S 的软件开发过程变得更加简单快捷。

STM8S 主要应用:

- 汽车电子领域:STM8S 应用于汽车传感器、执行器、安全系统控制、直流电机控制、车身电子控制、车载收音机、LIN 网络节点以及加热和空调系统。
- 工业应用领域:在家用电器、家庭自动化、电机驱动、空调系统、感应器、计量仪表、不间断电源(UPS)和安全监控等方面发挥着重要作用。
- 消费电子领域:STM8S 在电源管理、小型家用电器、音响设备、玩具、销售终端机、控制面板、电视以及监控设备中均有广泛应用。
- 医疗设备领域:STM8S 也被应用于个人医疗设备中,助力于提升医疗产品的功能性

和可靠性。

2. STM8L 超低功耗单片机

在 2009 年 9 月 15 日,意法半导体公司宣布其首批结合了高性能 8 位架构和最新超低功耗技术的 8 位微控制器——STM8L 系列,正式进入量产阶段。设计工程师可以利用 STM8L 系列来提升最终产品的性能和功能,同时满足市场对节能环保产品的需求。例如,便携设备、医疗设备、工业设备、电子计量设备以及感应或安全设备等,都需要延长电池的使用周期。STM8L 系列的超低功耗特性使其成为满足如“能源之星”、IEA 的“1W 节能计划”或欧盟 EuP 指令等低功耗设计标准的理想选择。

STM8L 系列的三个产品线都建立在意法半导体的超低功耗技术平台之上,该平台采用了公司独有的 130 纳米工艺,这一工艺特别优化了超低泄漏电流。技术亮点包括在整个 1.65V 到 3.6V 的电源电压范围内,CPU 能够以最大工作频率运行,实现最佳性能。此外,由于集成了一个片上稳压器,功耗与 V_{dd} 电压无关,这不仅提高了设计的灵活性,还有助于简化产品的设计过程。

STM8L 系列还包括了其他创新特性,如低功耗嵌入式非易失性存储器和多种电源管理模式:功耗仅为 5.4 μ A 的低功耗运行模式、3.3 μ A 的低功耗待机模式、1 μ A 的主动停止模式(实时时钟持续运行),以及功耗低至 350nA 的停止模式。STM8L 系列能够在 4 μ s 内从停止模式中快速唤醒,支持频繁使用最低功耗模式,以实现能效最大化。此外,低功耗外设,如功耗小于 1 μ A 的实时时钟和自动唤醒(AWU)模块,进一步助力节能。综合这些特性,STM8L 平台能够将动态电流消耗降至 150 μ A/MHz,展现了其在低功耗领域的领先地位。

STM8L 主要特点:

- 高效的处理能力:搭载 STM8 16 MHz CPU,提供快速的处理速度。
- 充足的存储空间:内置 4~32 KB 的闪存和高达 2KB 的 SRAM,满足不同规模的程序存储需求。
- 兼容性:三个产品系列之间实现了跨系列的引脚对引脚兼容、软件相互兼容以及外设相互兼容,简化了设计和迁移过程。
- 宽电压范围:电源电压覆盖 1.8~3.6 V,断电时最低可达 1.65 V,适应不同的电源环境。
- 超低功耗设计:在保持 SRAM 内容的超低功耗模式下,最低功耗仅为 350nA。
- 优化的运行功耗:运行模式下的动态功耗低至 150 μ A/MHz,实现能效优化。
- 先进的接口:集成了最先进的数字和模拟外设接口,增强了连接和数据处理能力。
- 宽工作温度范围:支持 -40 $^{\circ}$ C~+85 $^{\circ}$ C 的工作环境,部分产品可达 +125 $^{\circ}$ C,适用于多种环境条件。
- 易用性:提供免费的触感固件库,简化开发过程。

STM8L 主要应用:

- 医疗设备:适用于便携式医疗设备,支持医疗健康监测和数据管理。
- 娱乐领域:在玩具和游戏产品中,提供互动性和智能化功能。
- 交通管理:应用于公路收费系统,提高交通效率和自动化水平。
- 电池供电设备:为电池供电设备提供长效的电源管理和优化的能源使用。

- 个人保健产品:在个人保健产品中,实现健康数据的监测和分析。
- 安全系统:作为保安传感器的控制单元,增强家庭和商业环境的安全监控。

3. STM8A 应用汽车单片机

STM8A 主要特点:

- 集成的真数据 EEPROM:提供数据存储的可靠性。
- 双 RC 振荡器:内置 16MHz 的主振荡器和 128kHz 的低速振荡器。
- 高效的 STM8 内核:在 16MHz 频率下,能够达到 10MIPS 的性能。
- 高应用安全性:包括独立看门狗定时器和时钟安全系统,确保应用的稳定性。
- 全面的 LIN 2.0 支持:所有产品都具备 LIN 2.0 和自同步功能,便于实现汽车网络通信。
- 宽电源电压范围:适应 3.3V 和 5V 的电源电压。
- 高工作温度:能够承受高达 145°C 的工作温度。

STM8A 主要应用:

- 汽车激励器:用于控制汽车的动力输出。
- 车体控制器:管理车辆的各种电子系统。
- 传感器:监测汽车的运行状态和环境条件。
- DC 电机控制:用于汽车中直流电机的精确控制。
- 安全微控制器:在安全气囊等关键安全系统中发挥作用。
- LIN 节点:支持汽车中的局部互连网络通信。
- 汽车无线电:用于汽车娱乐和信息系统。
- HVAC 系统:控制汽车的加热、通风和空调系统

1.4 学习效果测评

1.4.1 测试题

【1】名词解释

- (1) 算术逻辑单元
- (2) 单片机
- (3) 总线
- (4) 时钟周期
- (5) 指令周期
- (6) 随机读写存储器
- (7) 操作数
- (8) 嵌入式系统

(9) 单片机哈佛架构

【2】简答题

- (1) 简述存储器写操作的具体步骤。
- (2) 简述单片机中指令执行的过程。
- (3) 嵌入式系统由哪几部分组成以及具有哪些特点?
- (4) 简述 STM8 系列单片机的分类。
- (5) 简述 STM8S 系列 MCU 的主要特点及应用领域。

【3】判断题

- (1) RAM(随机存取存储器)是计算机的永久存储设备。 ()
- (2) 数据总线是单向的,用于 CPU 与存储器、CPU 与外设或外设与外设之间传输信息。 ()
- (3) 机器周期是计算机执行最基本操作所需的时间长度,它由整数倍的时钟周期构成。 ()
- (4) 地址寄存器用于临时存储从存储器中读取的指令代码的第一个字节。 ()
- (5) 存储器主要分为两大类:只读存储器和随机读写存储器。 ()
- (6) 存储器芯片中存储单元的数量与地址线数目有关。 ()

【4】选择题

- (1) 计算机的中央处理器(CPU)主要负责:()
 - A. 数据存储
 - B. 数据输入
 - C. 数据处理
 - D. 数据输出
- (2) 下列哪一项不是计算机的输入设备?()
 - A. 键盘
 - B. 鼠标
 - C. 打印机
 - D. 扫描仪
- (3) 在计算机科学中,“算法”是指:()
 - A. 一种编程语言
 - B. 一组解决问题的步骤
 - C. 一种数据存储方式
 - D. 一种硬件组件
- (4) 在 CPU 中,哪个部件负责执行算术和逻辑运算?()
 - A. 寄存器
 - B. 控制单元
 - C. 算术逻辑单元
 - D. 总线
- (5) 在 CPU 中,哪个部件负责从内存中获取指令?()
 - A. 算术逻辑单元(ALU)
 - B. 控制单元(CU)
 - C. 指令寄存器
 - D. 数据总线
- (6) 嵌入式系统通常用于:()
 - A. 通用计算任务
 - B. 特定的控制任务
 - C. 网络服务器
 - D. 图形设计

(7)STM8 系列单片机通常使用哪种类型的编程语言? ()

A. Python

B. Java

C. C

D. SQL

1.4.2 评估表

本章重要知识点/技能点	掌握程度	已掌握内容	待加强内容	学习反思
计算机的基本组成				
CPU 的内部结构				
指令系统工作原理				
嵌入式系统组成与特点				
STM8 系列单片机分类及特点				
STM8 单品机结构				

第 2 章

STM8S 系列芯片内部结构

【主要知识内容】

STM8S 系列单片机是一种广泛应用于嵌入式系统的微控制器,其最小系统和结构设计对于确保单片机正常工作至关重要。这章主要介绍 STM8S 系列 MCU 的主要性能特点,如处理速度、内存大小、功耗等,以及 STM8S 系列 MCU 的内部组件如何协同工作。另外,还介绍 STM8S 系列的中央处理单元(CPU)的架构和功能;MCU 的物理封装类型和引脚的布局,I/O 引脚的物理结构,如何通过寄存器控制 I/O 端口的数据和行为,还包括电源管理,复位电和时钟电路等详细说明。通过本章的学习,对 STM8S 单片机会有一个系统的认识,为接下来学好其他章内容打好基础。

【学习目标】

理解 STM8S 系列微控制器的基本特性和应用领域;

熟练掌握各引脚功能;

熟练掌握 STM8S 系列芯片的内部结构,包括 CPU、存储器、I/O 端口和其他外设等。

【重点与难点】

重点:掌握通用 I/O 端口的内部结构及相关配置,复位电路寄存器的配置,时钟电路工作原理,供电系统功能及要求等。

难点:掌握 STM8S 系列芯片的内部结构。

【教学设计与实施方法】

本单元主要采用讲授教学法和练习教学法。讲授教学法通过教师课堂讲授实施,练习,教学法通过学生课后作业完成。

【实践环节设计】

本单元涉及 STM8 单片机最小系统等内容。

【教学效果测评】

通过提问、课后习题、项目实训等方式进行效果测评。

STM8 系列微控制器(MCU)由三个不同的产品线组成:STM8S(标准系列,工作电压为 3.0 至 5.0 伏特)、STM8L(低电压、低功耗系列,工作电压为 1.8 至 3.6 伏特)和 STM8A(专为汽车应用设计,工作电压为 3.0 至 5.0 伏特)。这些系列采用 0.13 微米工艺技术制造,集成了 MCU 领域近年来的多项创新技术,具有出色的性价比。STM8 内核 CPU 采用复杂指令集计算(CISC)架构,指令长度介于 1 到 5 字节之间,为程序设计提供了灵活性。每个机器周期对应一个时钟周期,CPU 的最高时钟频率可达 24 兆赫兹,大多数指令在 1 到 4 个机器周期内完成(除法指令除外),其中许多指令仅需 1 到 2 个机器周期即可执行,显示出快速的处理能力。该内核的指令集与 ST 公司早期的 ST7 系列产品基本一致,甚至兼容,内置了单线仿真接口模块,支持串行接口仿真(SIMM)方式,这有助于降低开发工具的成本。此外,STM8 系列 MCU 提供了丰富的内置外设,其中许多外设的内部结构和使用方法与基于 32 位 Cortex-M3 内核的 STM32 系列 MCU 相似或相同。STM8 系列 MCU 的功耗较低,功能全面,价格合理,性价比极高,适合应用于家用电器、电源控制与管理、电机控制、智能仪表等众多领域。它们是 8 位 MCU 控制系统升级换代的理想选择。

2.1 STM8S 系列 MCU 性能概述

STM8 内核 MCU 的主要特性如下:

(1)支持 16MB 线性地址空间:STM8 MCU 支持高达 16MB 的线性地址空间,将 RAM、EEPROM、Flash ROM 以及外设寄存器统一在一个地址空间内,确保了对所有类型的存储器和寄存器的访问指令格式一致,简化了编程过程。

(2)灵活的 I/O 引脚配置:I/O 引脚的设计允许用户根据外部电路的需求,通过编程选择引脚的工作模式,包括悬空输入、带内部上拉的输入、推挽输出和开漏输出等,极大地简化了外围电路设计。

(3)封装无关的引脚功能:不同封装的芯片之间,相同功能的引脚保持一致,没有交叉现象,这为硬件的扩展提供了便利。

(4)强大的抗干扰能力:每个输入引脚都配备了施密特触发器,增强了系统的抗干扰能力。

(5)高可靠性设计:STM8 MCU 配备了双看门狗(独立硬件看门狗和窗口看门狗),以及关键外设控制寄存器的原码和反码寄存器,提高了系统的稳定性。非法操作码检查功能在发现非法指令时会触发芯片复位,进一步确保了系统的可靠性。

(6)多样化的外设支持:STM8 MCU 提供了丰富的外设,包括可编程触发方式的外中断