

8098 单片微型计算机应用技术

李新民 李 勋 等 编著

北京航空航天大学出版社

(京)新登字 166 号

内 容 提 要

本书共十一章,在系统介绍 8098 片内各功能单元原理的基础上,以丰富的实例详细演示了中断系统、定时器、高速 I/O 口、串行口和 A/D、D/A 等环节的应用。这是本书的重要特色,对帮助读者正确使用 8098 有极大好处。

输入/输出和存储空间扩展一章除阐述了 8250、8279、8155、8255 和各种存储器的结构及原理外,还给出 8098 与它们的接口实例。

综合应用设计实例一章选材精练、内容新颖,概括了软硬件的主要应用技术。

本书还简要地叙述了 8098 的最新发展以及 1992 年后某些性能的变动情况。

主 编: 李新民 李 勋

编 著 者: (以撰写主要章节的先后为序)

李新民 李 勋 赵鸣生 林广艳

孙玉江 卢景山 李小京 张建友

●书 名: 8098 单片微型计算机应用技术

8098 DANPIAN WEIXING JISUANJI YINGYONG JISHU

●编 著 者: 李新民 李 勋 等

●责任编辑: 杨昌竹 杨远波

●出 版 者: 北京航空航天大学出版社(北京市海淀区学院路 37 号 100083)

●印 刷 者: 朝阳科普印刷厂

●发 行: 新华书店总店科技发行所

●经 售: 北京航空航天大学出版社发行科 全国各地新华书店

●开 本: 787×1092 1/16

●印 张: 15.75

●字 数: 403 千字

●印 数: 10100 册

●版 次: 1994 年 5 月第 1 版

●印 次: 1994 年 5 月第 1 次印刷

●书 号: ISBN-81012-482-X/TP·116

●定 价: 15 元

前 言

1990年,我们在北京航空航天大学出版社出版了《MCS-96系列8098单片微型计算机》一书。此后四年间,我们的许多年青教师都详细掌握了这种单片机的原理及应用知识。在此基础上,李新民副教授又组织并完成了若干个8098单片机的应用项目。本书正是这些实践的总结,故名《8098单片微型计算机应用技术》。

为照顾广大的初学读者,本书仍从8098单片机的原理讲起,并且每部分硬件资源的叙述都伴有大量的试验和应用实例。这不仅丰富了本书的内容,而且还大大增强了其可读性。

本书共分十一章。第一、二、三章由李新民撰写;第四、五、六、七章由赵鸣生和李勋撰写;第八章由林广艳、李新民撰写;第九章由林广艳和李勋撰写;第十章由林广艳、孙玉江、卢景山、李小京撰写;第十一章由张建友撰写。全书由李新民副教授主编,李勋教授主审并最后定稿。

在本书的编写过程中,我们得到了出版社和何立民教授的大力帮助,为此我们深表谢意。在审阅本书初稿时,何立民教授提出了许多宝贵意见。根据这些意见,我们对初稿进行了全面订正和补充。

我们将非常高兴地愿意看到各界读者对本书的批评意见。

联系地址:300160 天津纺织工学院计算中心。

编者著

1993年10月

目 录

第一章 8098 硬件结构概述	(1)
1.1 8098 的主要性能	(1)
1.2 封装及引脚功能	(2)
1.3 中央处理器(CPU)	(4)
1.3.1 CPU 总线	(4)
1.3.2 CPU 寄存器组合	(5)
1.3.3 寄存器算术逻辑单元 RALU	(5)
1.4 内部定时	(5)
1.5 存储空间	(6)
1.5.1 片内 RAM 空间	(6)
1.5.2 保留存储空间	(9)
1.5.3 片内 ROM 和 EPROM 空间	(10)
1.5.4 存储控制器	(10)
1.6 系统总线	(10)
1.6.1 总线时序	(10)
1.6.2 芯片配置寄存器 CCR	(11)
1.6.3 总线控制	(12)
1.7 就绪控制	(13)
1.8 程序封锁	(14)
1.9 I/O 口	(15)
1.10 复位	(15)
1.10.1 复位信号	(15)
1.10.2 复位状态	(15)
1.11 8×C198 和 8×C194	(16)
第二章 MCS-96 指令系统	(19)
2.1 操作数类型	(19)
2.1.1 字节型	(19)
2.1.2 字型	(19)

2.1.3	短整数型	(19)
2.1.4	整数型	(20)
2.1.5	位型	(20)
2.1.6	双字型	(20)
2.1.7	长整数型	(20)
2.2	操作数的寻址	(20)
2.2.1	寄存器直接寻址	(21)
2.2.2	间接寻址	(21)
2.2.3	自动增量间接寻址	(21)
2.2.4	立即寻址	(21)
2.2.5	短变址寻址	(22)
2.2.6	长变址寻址	(22)
2.2.7	零寄存器寻址	(22)
2.2.8	栈指针寄存器寻址	(22)
2.2.9	汇编程序对各寻址方式的选择	(22)
2.3	程序状态字	(23)
2.3.1	中断控制位	(23)
2.3.2	条件标志	(23)
2.4	指令系统概要	(24)
2.5	指令分类详解	(27)
2.5.1	算术指令	(27)
2.5.2	逻辑指令	(33)
2.5.3	数据传送指令	(35)
2.5.4	堆栈操作指令	(37)
2.5.5	跳转和调用指令	(38)
2.5.6	条件跳转指令	(39)
2.5.7	位测试并跳转指令	(40)
2.5.8	循环控制指令	(41)
2.5.9	单寄存器指令	(41)
2.5.10	移位指令	(42)
2.5.11	专用控制指令	(43)
2.5.12	规格化指令	(43)

第三章 中断系统..... (44)

3.1	中断源	(44)
3.2	中断控制	(46)
3.2.1	中断挂号寄存器	(46)
3.2.2	中断屏蔽寄存器	(47)
3.2.3	总体中断开关	(47)

3.2.4	中断优先级	(47)
3.3	中断响应过程的时间概念	(48)
3.4	中断系统实用技巧	(48)
3.4.1	中断嵌套	(49)
3.4.2	禁止中断嵌套	(51)
3.4.3	软件代请中断	(52)
3.4.4	非常规返回	(53)
第四章	定时器	(55)
4.1	定时器 1	(55)
4.2	定时器 2	(56)
4.3	定时器中断	(56)
4.4	监视定时器	(58)
4.5	软件保护措施	(58)
4.6	定时器应用实例	(59)
第五章	高速输入单元 HSI	(63)
5.1	HSI 的基本结构	(63)
5.1.1	硬件结构	(63)
5.1.2	HSI 输入引脚控制	(64)
5.2	HSI 的工作方式及其 FIFO 的运作	(65)
5.3	HSI 的工作状态	(66)
5.4	HSI 中断	(67)
5.4.1	HSI.0 中断	(67)
5.4.2	HSI 数据有效中断	(67)
5.5	HSI 应用实例	(67)
第六章	高速输出单元 HSO	(71)
6.1	HSO 硬件结构	(71)
6.2	HSO 的控制	(71)
6.3	HSO 状态	(73)
6.4	HSO 挂号的撤消	(73)
6.5	定时器 2 作 HSO 时基	(73)
6.6	软件定时器和 HSO 中断	(74)
6.7	HSO 应用实例	(75)
第七章	串行口	(85)
7.1	串行口的工作方式	(85)

7.1.1	方式 0	(85)
7.1.2	方式 1	(86)
7.1.3	方式 2	(86)
7.1.4	方式 3	(87)
7.2	串行口的控制	(87)
7.2.1	串行口控制和状态寄存器	(87)
7.2.2	串行口波特率的确定	(88)
7.2.3	多机通信	(89)
7.3	串行口的应用	(89)
7.3.1	方式 0 的应用	(89)
7.3.2	方式 1 的应用	(92)
7.3.3	方式 2 和方式 3 的应用	(94)
7.4	软件串行口	(100)

第八章 A/D D/A 模拟接口 (108)

8.1	模拟输入	(108)
8.1.1	A/D 转换命令	(109)
8.1.2	A/D 转换结果	(110)
8.1.3	用 HSO 起动 A/D 转换	(110)
8.1.4	A/D 转换器的接口	(111)
8.1.5	参考电压	(113)
8.1.6	提高分辨率的措施	(113)
8.2	确保 8098 A/D 采集精度的几个问题	(114)
8.2.1	模拟基准源的稳定	(114)
8.2.2	模拟量的稳定	(114)
8.2.3	应用系统的稳定	(114)
8.3	标度变换原理	(115)
8.3.1	线标变换公式	(115)
8.3.2	公式转换法	(116)
8.3.3	多项式变换公式	(116)
8.4	8098 A/D 转换死区问题	(116)
8.5	脉冲宽度调制输出 PWM	(118)
8.6	模拟输出	(119)
8.7	模入/模出应用程序实例	(120)

第九章 输入/输出及存储空间扩展 (126)

9.1	串行通信接口扩展	(126)
9.1.1	INS 8250 引脚功能	(127)

9.1.2	8250 内部寄存器	(130)
9.1.3	INS 8250 应用举例	(135)
9.2	并行 I/O 口 8155 扩展	(137)
9.2.1	结构和引脚功能	(137)
9.2.2	I/O 部分	(138)
9.2.3	8155 定时器	(140)
9.2.4	8098 与 8155 的接口	(141)
9.3	并行 I/O 口 8255 扩展	(144)
9.3.1	8255 的结构和功能	(144)
9.3.2	8255 的运行方式	(146)
9.3.3	8098 与 8255 的接口	(150)
9.4	8279 的键盘显示扩展	(151)
9.4.1	8279 功能概述	(152)
9.4.2	8279 引脚功能	(152)
9.4.3	8279 内部结构	(154)
9.4.4	8279 的命令格式与命令字	(155)
9.4.5	各种方式详述	(158)
9.4.6	8279 与 8098 接口	(161)
9.5	存储器扩展	(164)
9.5.1	8098 与 2764 连接	(165)
9.5.2	8098 与 87C257 连接	(165)
9.6	8098 与 E ² PROM 的连接	(165)
9.6.1	2864A 的运行方式	(165)
9.6.2	8098 与 2864A 接口	(167)
9.7	8098 与 RAM 的连接	(168)

第十章 综合应用设计实例

10.1	多单片机信息共享技术	(169)
10.1.1	概述	(169)
10.1.2	信箱存储系统	(169)
10.1.3	存储器共享技术	(173)
10.2	温度监测系统实例	(183)
10.2.1	概述	(183)
10.2.2	系统工作任务	(183)
10.2.3	系统硬件设计	(184)
10.2.4	系统软件设计	(190)
10.2.5	系统接地方法	(204)
10.2.6	系统接地设计实例	(205)
10.3	智能仪表的设计	(206)

10.3.1	设计思想	(206)
10.3.2	设计原理及实现方法	(207)
10.4	通用 8098 用户系统板的设计	(212)
10.4.1	总体结构	(212)
10.4.2	8098 用户系统板键盘及显示管理系统	(212)
10.4.3	8255 扩展的 I/O 口	(217)
第十一章 MCS-96/98 实用程序		(221)
11.1	数制转换	(221)
11.1.1	双字节整数十翻二子程序 ID2B	(222)
11.1.2	双字节小数十翻二子程序 PD2B	(222)
11.1.3	双字节整数二翻十子程序 I2BD	(223)
11.1.4	双字节小数二翻十子程序 P2BD	(223)
11.1.5	四字节整数十翻二子程序 ID4B	(223)
11.1.6	四字节小数十翻二子程序 PD4B	(224)
11.1.7	四字节整数二翻十子程序 I4BD	(225)
11.1.8	四字节小数二翻十子程序 P4BD	(225)
11.1.9	ASCII 码十进制输入转换成四字节二进制整数子程序 ASCDB	(226)
11.1.10	ASCII 码十进制输入转换成定点混合小数子程序 ASCSB	(227)
11.1.11	定点双字节混合小数二翻十子程序 B2ASC	(228)
11.1.12	ASCII 码转换成十六进制子程序 ASC16	(229)
11.1.13	一字节十六进制数转换成二位 ASCII 码子程序 HEXASC	(229)
11.1.14	三字节浮点数二翻十子程序 FBTD3	(230)
11.1.15	三字节浮点数十翻二子程序 FDTE3	(232)
11.2	数据处理	(234)
11.2.1	查表与散转	(234)
11.2.2	链表	(237)
11.2.3	队列	(239)
参考文献		(241)

第一章

8098 硬件结构概述

1.1 8098 的主要性能

16 位微控制器 MCS-96 家族成员众多,它们都擅长于高速控制功能。其新成员 8098 系专为要求 16 位微控制器速度而又局限于 8 位外部总线的应用场合设计的。该家族诸成员的共同特点是其 CPU 中均含有一个 256 字节的寄存器算术逻辑单元 RALU。8×98 包括统称为 8098 的下列三种器件:

- 8398 ——掩膜 ROM(8K 字节)型器件;
- 8798BH——EPROM(8K 字节)型器件;
- 8098 ——无片内 ROM 型器件。

8098 具有以下主要性能:

1. 17 位算术逻辑单元可对与其配合的 256 个字节的片内寄存器组合直接进行操作。这些寄存器全都具有累加器的功能,从而提供了高速的数据处理和频繁的输入输出能力,消除了累加器的瓶颈效应。

2. 高效的指令系统

8098 可进行高速算术运算,16 位加法只需 $1\mu\text{s}$ 即可完成。16 位乘法和 32 位对 16 位的除法也只不过用 $6.25\mu\text{s}$ 。这是 MCS-51 无法与之相比的。

3. 四通道 10 位 A/D 转换器,当晶振为 12MHz 时,A/D 转换时间为 $22\mu\text{s}$ 。

4. 可编程高速输入/输出机构 HSIO。高速输入器可用内部定时器 1 作实时时钟来记录外部事件发生的时间,一共可记录 8 个事件;高速输出可以按预定时间去触发某事件,并可根据需要挂号 1~8 个事件。

所谓高速,是指一旦对高速输入/输出机构编程后,它便可自动完成上述功能而无需 CPU 的干预。当晶振为 12MHz 时,其测量或产生脉冲的分辨率为 $2\mu\text{s}$ 。

5. 八个中断类型包括有 20 个中断源。每个中断类型分别对应各自的中断向量。

6. 全双工同步/异步串行口。与 MCS-51 系列单片机一样具有 4 种操作方式,可方便地实现 I/O 扩展,多机通信及与 CRT 终端等设备进行通信等。

7. 可用以作为 D/A 转换的脉冲调制输出 PWM。它可以直接驱动某些电机,也可以经过外部积分电路作直流输出。D/A 转换的分辨率为 8 位。当采用 12MHz 晶振时,其脉冲周期为 $64\mu\text{s}$ 。

8. 十六位监视定时器 WDT。当用户系统的软、硬件发生故障时,WDT 可使用户系统复

位,重新启动用户程序。

9. 两个 16 位定时器。定时器 1 在系统中作为标准时钟,不停地对内部时钟脉冲进行循环计数。定时器 2 主要用于对外部事件计数。

10. 四个软件定时器。它们均受高速输出机构 HSO 的控制,通过程序可使 HSO 在预定时间产生中断。每当预定的时刻已到,HSO 单元便把软件定时器标志置 1,并触发软件定时器的中断。

11. 芯片配置寄存器 CCR。8098 可以通过 CCR 的设置对总线控制信号的定义进行选择。并对就绪控制安排有若干种运行方式,从而提高了总线的灵活性,减轻访问慢速器件时对片外硬件的要求。

上述性能使 8098 成为 8 位微控制器工业中无与伦比的高级产品。

1.2 封装及引脚功能

8098 采用 48 引脚双列直插式封装形式(图 1-1)。

8098 的引脚功能如下:

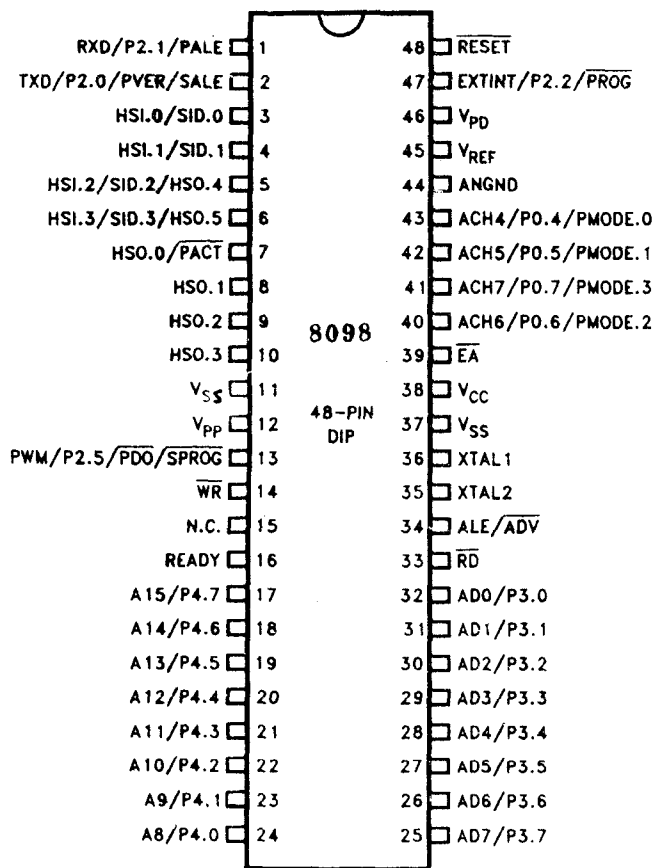


图 1-1 8098 引脚图

V_{CC} :主电源(5V)。

V_{SS} :数字电路地(0V)。有两个 V_{SS} 引脚,两者均必须连接。

V_{PD} :RAM 维持电源(5V)。正常运行时此电源必须接通,在掉电条件下(V_{CC} 降至 0),若在 V_{CC} 尚未降至规范值以下,但 V_{PD} 正常供电时, $\overline{RESET}$ 信号起作用,则寄存器组合顶端的 16 个字节将保持其内容不变。在掉电期间 $\overline{RESET}$ 必须保持低电平;在 V_{CC} 恢复正常,振荡器工作稳定之前, $\overline{RESET}$ 不应被拉成高电平。

V_{REF} :A/D 转换器参考电压(5V)。 V_{REF} 也是 A/D 转换器模拟部分及读 P0 口所用逻辑的供电电压。因此,为保证 A/D 和 P0 口正常工作, V_{REF} 必须接通。

ANGND:A/D 转换器参考地电平。通常必须和 V_{SS} 保持同电位。

V_{PP} :EPROM 型芯片的编程电压。

XTAL1:反相振荡器和内部时钟发生器的输入。

XTAL2:反相振荡器的输出。

$\overline{RESET}$:芯片的复位输入。两个状态周期以上的低电平输入将使芯片复位。随之而来的低至高的正跳变使 CLKOUT 再同步,并开始 10 个状态周期的内部复位序列。在此期间 PSW 被清 0,读自 2018H 单元的字节装入 CCR,最后控制 PC 转向 2080H 单元,正常运行时,此脚应输入高电平。 $\overline{RESET}$ 引脚具有内部上拉电阻。

$\overline{EA}$:存储器选择输入(片外访问)。 $\overline{EA}$ 为 1 将选中片内存储器的 2000H~3FFFH 单元;为 0,则选中片外程序存储器的这些单元。 $\overline{EA}$ 具有内部下拉电阻,故除非外部将其拉高,否则将处于低电平状态。 $\overline{EA}$ 的输入状态在复位期间被锁入内部。

$ALE/\overline{ADV}$:地址锁存允许或地址有效输出,由 CCR 选择。该引脚的两种功能均给出信号令锁存器把来自地址/数据总线的地址码锁存起来。当它作为 $\overline{ADV}$ 时,在总线周期的末尾升为高电平无效状态。 $\overline{ADV}$ 可用作片外存储器的片选信号。 $ALE/\overline{ADV}$ 仅在访问片外存储器期间起作用。

$\overline{RD}$:对片外存储器的读信号,输出,低电平有效。 $\overline{RD}$ 仅在读片外存储器期间起作用。

$\overline{WR}$:对片外存储器的写信号,输出,低电平有效。 $\overline{WR}$ 仅在片外存储器写操作期间起作用。

READY:延长片外存储器周期的就绪控制信号,输入,高电平有效,用以和慢速或动态存储器接口,或总线共享。总线周期可延长 $1\mu s$ 。当不用片外存储器时,READY 不起作用。插入总线周期的等待状态数由 CCR 控制。READY 内有弱上拉电阻。因此,除非从外部将其拉低,否则即为高电平。

HSI:高速输入单元的输入线,包括 HSI.0、HSI.1、HSI.2 和 HSI.3。其中 HSI.2 和 HSI.3 与 HSO 单元共用。

HSO:高速输出单元的输出线,有 HSO.0、HSO.1、HSO.2、HSO.3、HSO.4 和 HSO.5 六条线。其中 HSO.4 和 HSO.5 与 HSI 共用。

P0 口:4 位高阻输入端口。这些引脚可作数字输入,也可作 A/D 转换器的模拟输入。

P2 口:4 位多功能口。其引脚与 8098 其他功能共用。

P3 和 P4 口:8 位双向 I/O 口,漏极开路输出。这些引脚和多路切换的地址/数据总线共用,具有内部强上拉电阻。

1.3 中央处理器(CPU)

图 1-2 为 8098 结构的简化框图。8098 CPU 主要包括快速寄存器组合,专用寄存器 SFR,存储控制器和寄存器/算术逻辑单元 RALU。与外部世界的通信是通过 SFR 或存储控制器进行的。RALU 并不使用累加器,而直接对寄存器组合及诸专用寄存器构成的 256 字节地址空间进行操作。CPU 通过专用寄存器直接控制 I/O,可使 I/O、A/D、PWM 和串行口的工作效率大为提高。这种结构的主要优点在于具有上、下文切换的能力,无累加器不足的困扰,以及数据

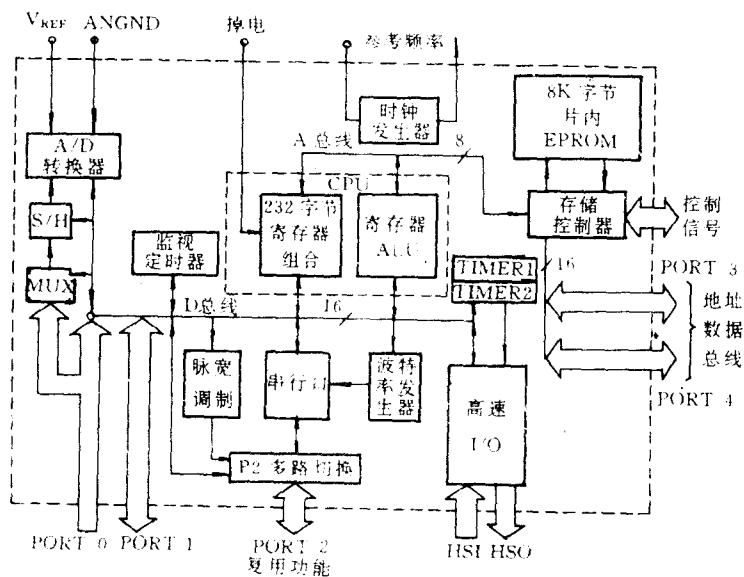


图 1-2 8098 结构框图

吞吐和 I/O 的快速性等等。

1.3.1 CPU 总线

由图 1-2 可见,CPU 的两个主要组成部件——RALU 和寄存器组合——是通过控制单元和两组总线连接起来的。此两组总线中,A 总线宽 8 位,D 总线宽 16 位。D 总线只负责 RALU 和寄存器组合(或专用寄存器)间的数据传送。A 总线则用作上述传送的地址总线,或作为连接存储控制器的地址/数据多用总线。必须强调指出,对片内 ROM 或片外存储器的访问都是通过存储控制器进行的。

存储控制器内部有一从程序计数器(从 PC)负责跟踪 CPU 的主 PC。程序的大多数取指令操作均由从 PC 指挥,几乎无需把地址送往存储控制器,这样可使处理器节约时间。若执行转移指令,则要有新的地址装入从 PC 中,并由此继续执行程序。由存储器中取数据也要经过存储控制器,但这种操作是不通过从 PC 的。

对 RALU 的指令由存储控制器负责从 ROM 中取来,并经 A 总线送至指令寄存器。此后

控制单元便对所取指令进行译码并产生相应序列的信号以使 RALU 完成所要求的功能操作。

1.3.2 CPU 寄存器组合

寄存器组合包括 232 个字节的 RAM, 对它们可按字节、按字或按双字进行寻址。因为其中任一单元均可作为 RALU 所用, 所以, 实质上有 232 个“累加器”。寄存器组合中的第一个字用作堆栈指针, 故当有堆栈操作时, 该字不得用以存放数据。访问寄存器组合和专用寄存器时, 其地址由 CPU 硬件暂存在两个 8 位地址寄存器中。

1.3.3 寄存器算术逻辑单元 RALU

8098 的大多数运算都是在 RALU 中进行的。RALU 的结构如图 1-3 所示, 它含有一个 17 位 ALU、程序状态字 PSW、程序计数器 PC、循环计数器以及三个暂存寄存器。上述所有寄存器均为 16 位或 17 位(16 位加符号扩展位)。其中有些寄存器具有简单的运算能力, 以减轻 CPU 的负担。

PC 附有一个专用增 1 器, 用以在每取指令一个字节后使 PC 值自动增 1; 然而跳转指令必须经过 ALU 处理。三个暂存寄存器中, 有两个自身具备移位逻辑。它们用来进行某些需要逻辑移位(包括规格化、乘、除运算)的操作。“低字”寄存器仅仅在双字移位时才用到; “高字”寄存器只要进行移位就要用到, 但也可作为许多指令作暂存寄存器用。重复移位由 5 位循环计数器控制。

第三个暂存寄存器的任务是在执行双操作数指令时存放第二个操作数。这包括乘法运算的乘数和除法运算的除数。在做减法时, 此寄存器的输出可在送入 ALU 的 B 输入端前先行取反。

图 1-3 中的延时环节用来把 16 位总线变成 8 位总线, 即把 ALU 同时送出的 16 位数分成高低 8 位先后送往 8 位 A 总线。这一做法也是在 8 位 A 总线上传送所有地址和指令时所要求的。在 RALU 中也存有若干常数, 如 0、1、2 等, 它们的任务在于加速某些运算。譬如, 当 RALU 需要求补或执行增 1、减 1 指令时, 这些常数就会有用。

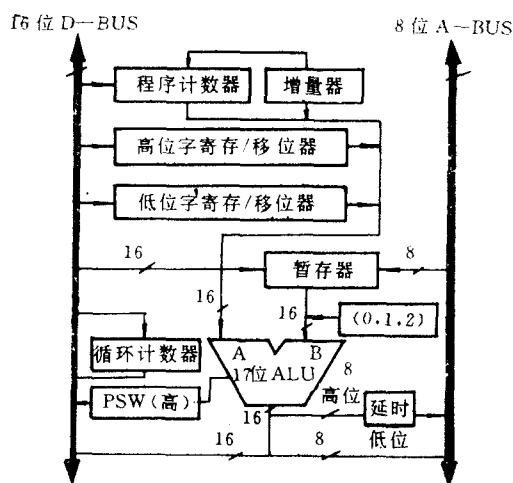


图 1-3 RALU 框图

1.4 内部定时

8098 要有 6~12MHz 间的输入时钟频率才能正常发挥其功能。此频率可直接加至 XTAL1。另一方案是, 可用一晶体来产生时钟信号, 因为 XTAL1 和 XTAL2 分别为一反相器的输入和输出。图 1-4 所示为振荡部件的框图。

晶体或外部振荡频率被三分频后便得到三相内部定时信号(图 1-5)。各相每过三个振荡周期重复一次, 故三个振荡周期被称为一个状态周期——8098 操作的基本时间单位。若振荡

频率为 12MHz, 一个状态周期为 $0.25\mu\text{s}$, 8 个状态周期为 $2\mu\text{s}$ 。

大部分内部操作或与 A 相、B 相或与 C 相同步。三相信号的占空比均为 33%。A 相在外部被标作 CLKOUT, 这是 68 引脚的 8096 的一个时钟输出信号。B 相和 C 相并不对外。XTAL1、CLKOUT 和 A、B、C 三相信号间的关系见图 1-5。应当指出, 图 1-5 没有给出传递延时。

RESET 信号线可用来启动 8098, 向测试设备和多芯片系统提供精确的同步信号。

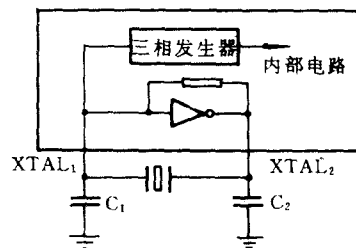


图 1-4 振荡器框图

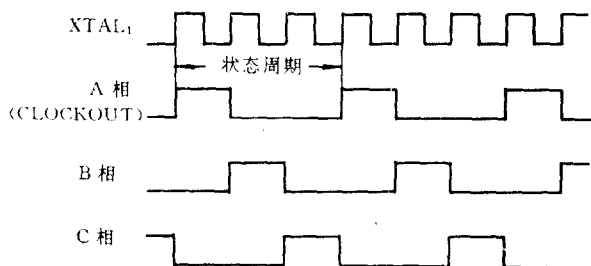


图 1-5 相对 XTAL1 的内部定时

1.5 存储空间

8098 的可寻址空间为 64K 字节, 采用 Von Neuman 结构, 即程序与数据空间统一编址。其中 0000H~00FFH 以及由 1FFE H 至 207FH 为专用空间。此外所有单元均归用户分配, 可用来存放程序, 亦可用以存放数据, 抑或作为外部接口的存储映象。图 1-6 所示为存储空间分布情况。

1.5.1 片内 RAM 空间

00H 至 FFH 单元为寄存器组合和专用寄存器的空间。这一片内 RAM 区间的数据不可作指令代码去执行。若从 00H 至 FFH 单元取指令去执行, 则这些指令将取自片外存储器。片外存储器的这段地址空间留供开发工具所用。

1. 00H~17H 这 24 字节为专用寄存器空间。8098 的所有 I/O 均由专用寄存器控制。许多专用寄存器具有两种功能, 对其进行读或写时, 功能有所不同。图 1-6 中给出了这些寄存器的地址和名称。下面我们概括地介绍一下各专用寄存器的功能, 详细论述留待后续章节进行。

R0:0 寄存器, 读时恒为 0, 可作变址寻址的基地址, 亦可作计数和比较时的常数。

AD_RESULT: A/D 结果寄存器, 存放 A/D 转换结果的高、低字节。字节只读。

AD_COMMAND: A/D 命令寄存器, 用以控制 A/D 转换。

HSI_MODE: 高速输入方式寄存器, 负责设置高速输入的运行方式。

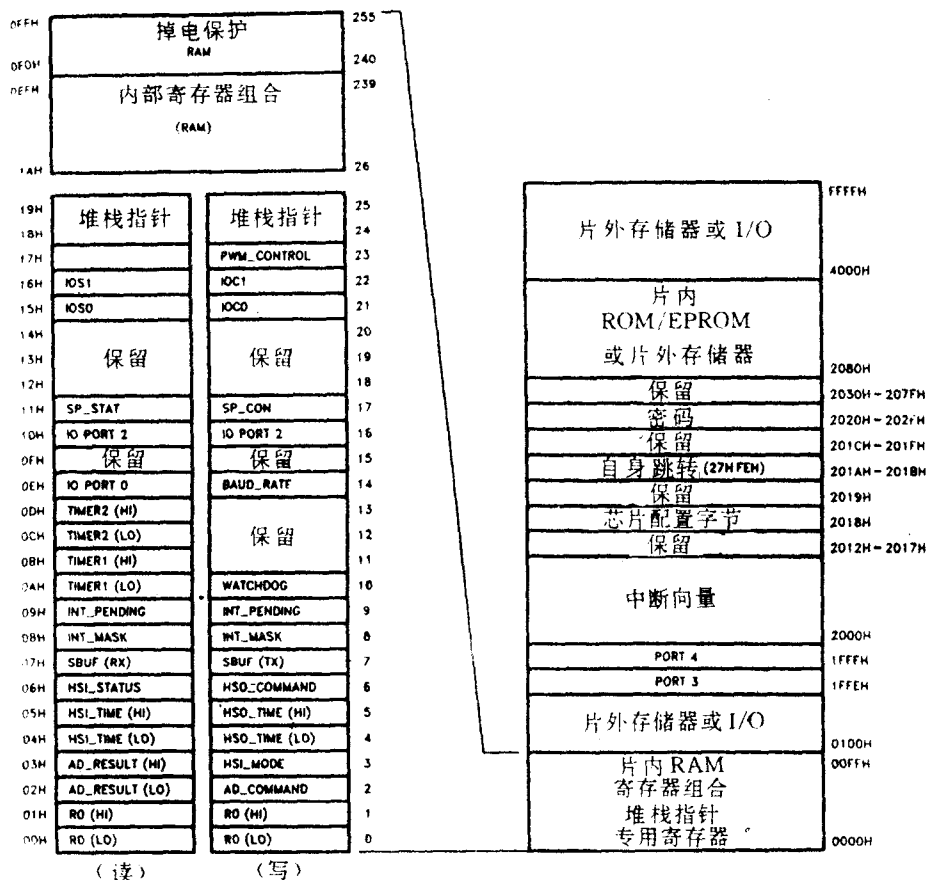


图 1-6 存储空间图

HSI_TIME: 高速输入时间寄存器, 存放 HSI 定时的高低字节, 用以记录高速输入单元的触发时间, 字只读。

HSO_TIME: 高速输出定时的高、低字节, 用以设定高速输出单元执行命令寄存器中命令的时间或次数, 字只写。

HSO_COMMAND: HSO 命令寄存器, 负责确定在装入 HSO_TIME 的时间到后该发生什么情况。

HSI_STATUS: HSI 状态寄存器, 负责指示在 HSI_TIME 寄存器所记录的時刻哪个 HSI 引脚上发生了事件以及这些高速输入引脚的现行状态。

SBUF(TX): 串行口发送缓冲器, 用以保存要发送的内容。

SBUF(RX): 串行接收缓冲器, 其任务在于保存刚刚收到的字节。

INT_MASK: 中断屏蔽寄存器, 用以单独开放或关闭各源之中断。

INT_PENDING: 中断挂号寄存器, 表示中断源中某中断请求有效, 但仍未获得响应。

WATCHDOG: 监视定时寄存器。程序正常运行时, 定期改写其内容使之永不溢出; 程序发生故障时, 监视定时器就会溢出, 使 8098 复位, 这样可以保证程序安全运行。

TIMER1: 定时器 1 的高、低字节, 字只读。

TIMER2: 定时器 2 的高、低字节, 字只读。

BAUD_RATE: 波特率寄存器, 字节只写, 读时为 P0 口的 4 位输入。

IO_PORT1: P1 口寄存器, 用于 P1 口的读写。这是 8096BH 的 P1 口, 8098 无此端口。

IO_PORT2: P2 口寄存器, 用于 P2 口的读写。

SP_STAT: 串行口状态寄存器, 指示串行口的状态。

SP_CON: 串行口控制寄存器, 用以设置串行口的运行方式。

IOS0: I/O 状态寄存器 0, 含有 HSO 的状态信息。

IOS1: I/O 状态寄存器 1, 含有 HSI 和定时器的状态信息。

IOC0: I/O 控制寄存器 0, 控制 HSI 引脚的多用功能、定时器 2 复位源及时钟源。

IOC1: I/O 控制寄存器 1, 控制 P2 口引脚的多用功能、定时器中断和 HSI 中断。

PWM_CONTROL: 脉宽调制控制寄存器, 设置 PWM 脉冲的持续时间。

2. 18H~19H 用作堆栈指针。这两个单元并非专用寄存器, 若无堆栈操作它们可作普通 RAM 单元来用。

3. 1AH~0EFH 可自由地作为通用寄存器使用。它们可以按字节、字或双字存取。

4. F0H~FFH 为片内掉电保护空间, 由 V_{PD} 引脚供电。若希望在断电期间保持这些单元的内容有效, 只需保持该引脚电压即可。保持该区内容不消失所需电流约为 1mA, V_{PD} 电压最低为 4.5V, 最高为 5.5V。故以电池作此引脚之后备电源亦无困难。不过应注意, 正常运行时 V_{CC} 和 V_{PD} 必须同时供电。

为使 8098 进入掉电方式, $\overline{RESET}$ 引脚应被拉成低电平。这样, 两个状态周期后, 芯片即处于复位状态。这一点之所以必要, 是因为要防止掉电过程中向 RAM 写入任何数据。此后, 电源便可从 V_{CC} 引脚撤消, 而 V_{PD} 则应保持在 4.5~5.5V 之间。8098 可无限期地停留在这种状态下, 同时上述 16 个 RAM 字节将维持其内容不变。

欲使 8098 退出掉电状态, 应在施加 V_{CC} 电压的同时, 保持 $\overline{RESET}$ 为低电平。当时钟电路启动时, 8098 应具有稳定的电源电压。时钟振荡稳定两个状态周期后 $\overline{RESET}$ 引脚即可被拉成高电平。拉高后再过 10 个状态周期, 8098 将从 2080H 单元开始执行程序。图 1-7 所示为掉电运行序列的时序图。为保证两个状态周期能满足最小复位时间(与 CLKOUT 同步)的要求, 建议

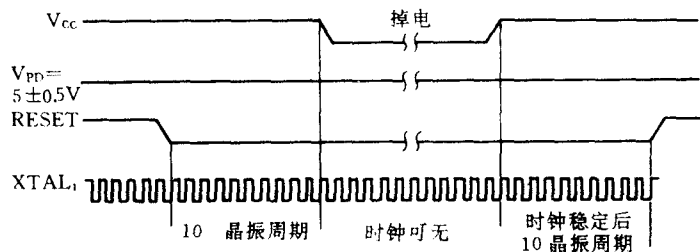


图 1-7 掉电前后的时序图