

Embedded in Life



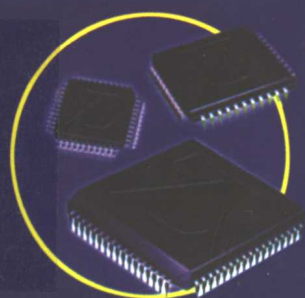
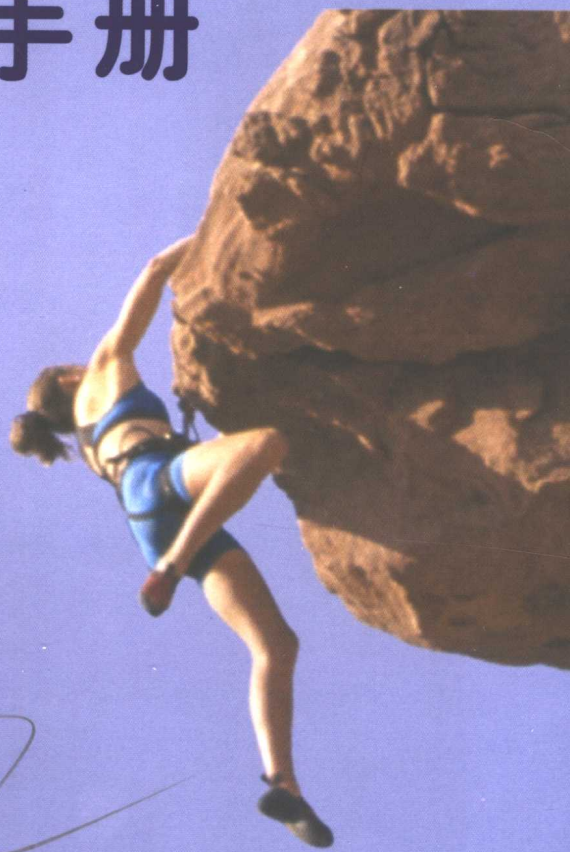
Z8 Encore!

八位微控制器原理、 应用及技术手册

董伯明 主编

黎秉志等 翻译

姚 萌 主审



复旦大学出版社

Z
Z i L O G

Z8 Encore! 八位微控制器原理、 应用及技术手册

董伯明	主编
黎秉志等	翻译
姚 萌	主审

復旦大學出版社

图书在版编目(CIP)数据

Z8 Encore! 八位微控制器原理、应用及技术手册/董伯明主编. —上海:复旦大学出版社,2005.1
ISBN 7-309-04320-0

I. Z... II. 董... III. 微控制器 - 技术手册
IV. TP332.3 - 62

中国版本图书馆 CIP 数据核字(2004)第 137117 号

Z8 Encore! 八位微控制器原理、应用及技术手册

董伯明 主编

出版发行 复旦大学出版社

上海市国权路 579 号 邮编:200433

86-21-65118853(发行部); 86-21-65109143(邮购)

fupnet@fudanpress.com <http://www.fudanpress.com>

责任编辑 陆盛强

总编辑 高若海

出品人 贺圣遂

印刷 句容市排印厂

开本 787 × 1092 1/16

印张 31.875

字数 856 千

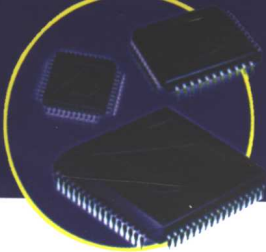
版次 2005 年 1 月第一版第一次印刷

书号 ISBN 7-309-04320-0/T · 291

定价 48.00 元

如有印装质量问题,请向复旦大学出版社发行部调换。

版权所有 侵权必究



Federico Faggin

Date of Birth: December 1st, 1941

Nationality: Italy



BRIEF BIOGRAPHY

- 1941 Born in Vicenza, Italy
- 1965 Doctor in physics, University of Padua, Italy
- 1967 SGS-Fairchild, Italy
- 1970 Intel Corp., joined the development project for 4004 microprocessor
- 1974 President, Zilog Inc., U.S.A.
- 1982 President, Cygnet Technologies, Inc.
- 1986-present President, Synaptics, Inc.

AWARDS AND HONORS

- 1988 International Marconi Fellowship Award, England
- Gold Medal for Science and Technology, Italy
- 1996 "25 Years of Industry Achievement, ZiLOG Z80" with Shima, M. and other, Fall Comdex, U.S.A.
- 1996 "The 1996 PC Magazine, Awards for Technical Excellence" with Hoff, M.E., Mazor, S. and Shima, M., Fall sComdex, U.S.A.

MAJOR WORKS

- 1972 "Standard Parts And Customs Design Merge In Four-Chip Processor Kit" with Hoff, M.E., Electronics
- 1972 "The MCS-4 An LSI Microcomputer System" with Hoff, M.E., Mazor, S., Shima, M. and other, IEEE
- 1975 "Trends In Microprocessors" IEEE
- 1976 "Z80: Chip Set Heralds Third Microprocessor Generation" with Shima, M. and other, Electronics
- 1978 "How VLSI Impacts Computer Architecture" IEEE Spectrum
- 1996 "The History of the 4004" with Hoff, M.E., Mazor, S. and Shima, M., IEEE Micro
- 1974 3,821,715 Memory System for Multi-Chip Digital Computer, with Hoff, M.E. and Mazor, S.

nce
by the
t
ne
to
st
s.
red
relop
ning
7
ks
d
It
the
o
y
a
to

序 言

我非常荣幸地为第一本中文版的 Encore! 技术丛书——最新的 Z8 单片机系列产品作序。

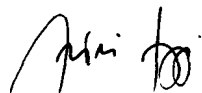
自从 1978 年第一个 Z8 单片机产品问世以来,Encore! 系列产品是代表了当今市场所需的主要升级版的 Z8 家族之一。

开始设想研发 Z8 可以追溯到 1974 年末,这是我所孕育的第一个产品。不久,我创建了 ZiLOG 公司,约一个月之后,我又开始了 Z80 中央处理单元的项目,就此决定先开发完成 Z80,然后再是 Z8。

多年来,好几代的设计人员应用 Z8 研发了成千上万的应用项目,其中很多项目甚至是我在 1974 年时都不可能想象的。

使我感到非常欣慰的是我亲眼看到 Z8 在市场上的巨大成功,并且深信在 Z8 家族中诞生发展了 Encore! 系列产品,更能确保 Z8 在新纪元会继续成为在世界上应用最广泛的单片机之一。

Encore! 万岁。



Federico Faggin

前 言

美国 ZiLOG 公司以其独具特色的 eZ8 核开发出带 Flash 和各种功能模块,诸如 10 位 ADC, 8 位 PWM, 内部时钟 IPO, 内带温度传感器等功能的嵌入式的 eZ8x 单片机系列产品,并具有与 Z8, Z8+ 的软件向下兼容性。

同时,开发出实时、多功能、方便、友好的开发仿真系统,售价仅 39.90 美元。

为方便中国的广大用户和有关方面成员的先睹为快,特选择了最相关的三篇内容率先译成中文出版,后续的还会陆续出版。

“单片机之父”Federico Faggin 还专门为此书写了序,在此附上了他的手稿与履历。

此书由西安石油大学的黎秉志教授、霍汉平、段纬林、王小燕、谢文昊等老师及华东师范大学的姚萌教授、刘丽力、王晓栋、郑事隽等翻译,并由姚萌教授担任主审。在此一并感谢。

需了解更详细、更新的 ZiLOG 信息,可查阅 ZiLOG 公司网站。

网址: www.zilog.com; www.zilog.com.cn

ZiLOG 公司中国区首席代表
董伯明

目 录

第一篇 eZ8 CPU 使用手册	1	六、查询中断处理	20
第一章 结构概述	3	七、软件中断的产生	20
一、特性	3	第六章 非法的指令陷阱	22
二、处理器定义	3	一、概述	22
三、取指令单元	3	二、一个非法指令陷阱的符号化	
四、指令状态器和指令周期	3	运算	22
五、程序计数器	4	三、不使用堆栈的线性程序	22
六、CPU 控制寄存器	4	第七章 eZ8 CPU 指令概述	23
第二章 eZ8 兼容性	7	一、汇编语言概述	23
一、概述	7	二、汇编语言语法	23
二、汇编语言兼容性	7	三、eZ8 CPU 指令记号法	24
三、新指令	7	四、eZ8 CPU 指令分类	25
四、eZ8 CPU 控制寄存器的新位置	8	五、eZ8 CPU 指令概述	27
五、堆栈指针兼容性	8	第八章 eZ8 CPU 指令系统说明	34
六、复位兼容性	8	第九章 操作码示意图	87
七、中断兼容性	9	第十章 操作码指令表	90
第三章 地址空间	10	第十一章 汇编程序和操作码	
一、概述	10	的例子	97
二、寄存器阵列	10	第二篇 带闪存和 10 位 A/D 的高性能	
三、程序存储器	12	单片机	105
四、数据存储器	13	第一章 概述	107
五、堆栈	13	一、特征	107
第四章 寻址方式	14	二、部件选择指南	107
一、概述	14	三、结构图表	107
二、寄存器寻址(R)	14	四、CPU 和外部设备概述	108
三、间接寄存器寻址(IR)	16	第二章 引脚配置	110
四、变址寻址(X)	16	一、概述	110
五、直接寻址(DA)	17	二、可用的封装	110
六、相对寻址(RA)	17	三、引脚配置	110
七、立即数寻址(IM)	18	四、信号描述	111
第五章 中断	19	第三章 地址空间	114
一、概述	19	一、概述	114
二、允许中断和禁止中断	19	二、寄存器阵列	114
三、中断优先级	19	三、程序存储器	114
四、矢量式中断处理	19	四、数据存储器	115
五、矢量中断的嵌套	20		

五、Flash 信息区	115	定义	160
六、寄存器映射	115	第十一章 通用异步收发器	162
第四章 运行	119	一、概述	162
一、CPU 与外设概述	119	二、结构	162
二、通用 I/O	119	三、操作	163
第五章 复位和停止模式恢复	121	四、UART 中断	167
一、概述	121	五、UART 控制寄存器定义	169
二、复位类型	121	第十二章 红外编码/解码	174
三、复位寄存器定义	125	一、概述	174
第六章 低电源模式	126	二、结构	174
一、概述	126	三、操作	174
二、停止模式	126	第十三章 ADC 模数转换器	177
三、中止模式	126	一、概述	177
四、外设级电源控制	127	二、结构	177
五、电源控制寄存器定义	127	三、操作	177
第七章 通用 I/O	128	四、可编程触发点警报	179
一、概述	128	五、中断	180
二、驱动 GPIO 端口的设备可用性	128	六、校准和补偿	180
三、系统结构	128	七、输入缓冲阶段	181
四、GPIO 变换功能	129	八、跨导倒数放大器	181
五、GPIO 中断	130	九、ADC 控制寄存器的定义	182
第八章 中断控制器	137	第十四章 比较器	186
一、概述	137	一、概述	186
二、中断向量列表	137	二、操作	186
三、体系结构	138	三、比较器控制寄存器的定义	186
四、操作	138	第十五章 温度传感器	188
五、中断控制寄存器定义	139	一、概述	188
第九章 定时器	145	二、操作	188
一、概述	145	三、校准	188
二、体系结构	145	四、校准后/平衡后的温度传递	
三、操作模式	145	函数表	188
四、读取计数器的值	153	五、调整细节	189
五、定时器引脚信号的操作	153	第十六章 闪存	190
六、计数器控制寄存器的定义	153	一、概述	190
第十章 看门狗定时器	158	二、Flash 信息区	191
一、概述	158	三、操作	191
二、操作	158	四、使用 Flash 频率寄存器进行	
三、看门狗定时器刷新	158	Flash 定时操作	191
四、在正常操作下的 WDT 中断	159	五、防止外部访问的 Flash 代码保护	191
五、看门狗定时器重载解锁顺序	159	六、防止突发程序写入或擦除的	
六、看门狗定时器控制寄存器的		Flash 代码保护	191

七、字节编程	193	四、片上外围交流/直流设备的电气特性	232
八、Flash 控制器的旁路设置	194	五、通用 I/O 口输入数据采样时序	235
九、Flash 控制寄存器的定义	194	六、通用 I/O 端口输出时序	236
十、Flash 选项位	196	七、片上调试器时序	236
第十七章 非易失数据存储	201	八、通用异步收发器(UART)时序	237
一、概述	201	第二十四章 封装	239
二、操作	201	第二十五章 命令信息	240
三、非易失数据存储(NVDS)代码接口	201	第二十六章 产品号的描述	243
第十八章 片上调试器	203	第三篇 Z8 Encore! 开发工具用户手册	
一、概述	203	——ZDSII	245
二、结构	203	第一章 概述	247
三、操作	203	一、系统配置	247
四、调试模式	203	二、软件安装	247
五、断点	205	三、Z8 Encore! 开发环境工具	247
六、运行时间计数器	205	四、开发环境快速教程	249
七、片上调试器命令	206	五、调试快速教程	254
八、片上调试器控制寄存器的定义	209	六、仿真器快速教程	258
第十九章 振荡器控制	211	七、周期精准仿真器快速教程	259
一、概述	211	八、命令处理器快速教程	259
二、操作	211	九、从命令行运行 ZDSII	277
三、时钟故障检测和恢复	212	第二章 主要组件	284
四、振荡器控制寄存器的定义	212	一、工具条	284
第二十章 晶体振荡器	214	二、窗口	287
一、概述	214	三、菜单栏	291
二、操作模式	214	第三章 调试	329
第二十一章 内部精准振荡器	216	一、状态条	329
一、概述	216	二、代码行数指示	329
二、操作	216	三、调试输出窗口	330
第二十二章 eZ8 CPU 指令集	217	四、调试工具条	330
一、汇编语言编程介绍	217	五、Debug 窗口	331
二、汇编语言源程序举例	217	六、使用断点	341
三、汇编语言语句	217	七、使用仿真器	343
四、eZ8 CPU 指令符号	218	第四章 ANSI C-Compiler	347
五、eZ8 CPU 指令分类	219	一、语言范围	347
六、eZ8 CPU 指令概要	221	二、使用实时库	349
七、操作码映射	227	三、实时库功能	362
第二十三章 电气特性	230	四、在 C 环境下设置 Flash Option bits	392
一、最大额定值	230		
二、直流特性	230		
三、交流特性	231		

五、通过#Pragmas 设置 C 编译器 选项	393	八、结构编译处理	420
六、在 C 中嵌入汇编指令	394	九、条件编译	421
七、在 C 语言中调用汇编	394	十、宏	423
八、C 语言编程中的常见问题	395	十一、标注	425
九、错误信息	396	十二、源程序语法	426
第五章 应用宏汇编	404	第六章 使用链接/定位器	436
一、地址空间和地址段	404	一、链接器命令	436
二、输出文件	405	二、链接器表达式	440
三、语言结构	406	三、链接器释疑	444
四、汇编程序规则	407	四、链接器警告信息	445
五、表达式	408	附录 A ANSI C 语言参考	446
六、操作优先权	410	附录 B ANSI C 预处理命令	483
七、汇编命令	410	附录 C 词汇表	490
		附录 D 部分产品封装	495

第一篇 eZ8 CPU 使用手册

第一章 结构概述

一、特 性

eZ8 是 ZiLOG 公司新近推出的 8 位中央处理单元(CPU)快速和高效编码的微控制器。eZ8 执行一系列源于 Z8 的指令设置。eZ8 的特性包括:

- 直接寄存器到寄存器访问的结构,每个寄存器可以当作累加器。这减少了指令的执行时间和程序存储空间。
- 对于子程序调用和中断响应,软件堆栈的深度比硬件堆栈更大。
- 与 Z8 汇编指令兼容。
- 内部的寄存器阵列扩展到 4KB。
- 新的指令通过使用高级编程语言(包括 C 语言)拥有更高的执行效率。
- 设计了取指令和执行指令的管道线。
- 流水线式命令读取和执行。

二、处理器定义

eZ8 CPU 包括两个主要的功能块——取指令单元和执行单元。执行单元还可以进一步分为指令状态器、程序计数器、CPU 控制寄存器和算术逻辑单元(ALU)。eZ8 CPU 的结构见图 1.1.1。

三、取指令单元

取指令单元用于控制存储器的接口。其最基本的功能是从存储器中读取操作码和操作数。

取指令单元可以解读部分操作码以决定对于本次操作的读取字节的数量。取指令单元的操作顺序如下:

1. 读取操作码; 2. 判断操作数大小(数或字节); 3. 读取操作数; 4. 发送操作码和操作数给指令状态器。

取指令单元是管道线式并且半独立地在 eZ8 CPU 中操作。

四、指令状态器和指令周期

指令状态器是 eZ8 CPU 执行单元的控制单元。在取指令单元中初始化操作码后,指令状态器会关注并执行指令。指令状态器读写寄存器并且产生地址。

指令与指令的指令周期是不同的,这使得单片机具有更高的性能,对标准的 CPU 指令来说,最小的指令执行周期只有两个时钟周期(只有 BRK 指令是单周期的)。由于每条指令需要改变的字节数是不同的,执行指令时将会有延迟周期,这条指令的延迟周期将会加到下一条指令的执行周期中。比如,如果 eZ8 CPU 在读取 3 字节的指令时要执行 2 周期的指令,因为读取单元只有两个周期去读取 3 字节,所以将发生延迟,执行单元在延迟周期处于空闲状态。

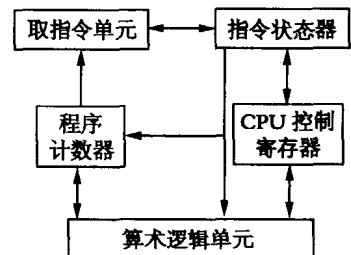


图 1.1.1 eZ8 CPU 结构图

五、程序计数器

程序计数器包括 16 位的计数器和 16 位的加法器。程序计数器监测着现在存储器的地址和下一个存储器的地址,程序计数器根据取指令单元中被读取的字节数的增加而自动增加。16 位的加法器在程序计数器跳转到相关地址时自动增加。

六、CPU 控制寄存器

eZ8 CPU 包含 4 个 CPU 控制寄存器,这些控制寄存器被映射入寄存器阵列的地址空间。这 4 个 eZ8 CPU 控制寄存器是:堆栈指针高字节、堆栈指针低字节、寄存器指针、标志寄存器。

eZ8 CPU 的寄存器总线有多达 4K 字节的寄存器空间。在所有 eZ8 CPU 产品中,前 256 字节是为控制 eZ8 CPU、片上调试和 I/O 口而预留的。eZ8 CPU 控制寄存器的地址被固定为 FFCH 到 FFFH。见表 1.1.1。

表 1.1.1 eZ8 CPU 控制寄存器

寄存器助记符	存储器说明	地址 (Hex)
FLAGS	标志寄存器	FFC
RP	寄存器指针	FFD
SPH	堆栈指针高字节	FFE
SPL	堆栈指针低字节	FFF

1. 堆栈指针寄存器

eZ8 CPU 允许用户在寄存器页重新定义堆栈。堆栈可以设置在地址 0000H 到 EFFFH,12 位的地址由 SPH[3:0]和 SPL[7:0]构成。堆栈指针对于堆栈的操作有 12 位的自增和自减功能,允许堆栈指针可以操作在不同的寄存器页,堆栈指针寄存器的值在复位之后没有定义。

2. 寄存器指针

寄存器指针包括当前工作的寄存器组和寄存器页的地址信息。页指针是寄存器指针的低 4 位,RP[3:0],指向当前页。一共有 16 页 256 字节的寄存器可供用户使用。工作寄存器组指针寄存器指针的高 4 位,RP[7:4],可以指向 16 个都具有 16 字节的工作寄存器组中的一个。每页有 16 个工作寄存器组。其他信息请见“地址空间”章节。

3. 标志寄存器

标志寄存器包括当前算法和逻辑状态信息。状态寄存器包括 6 个表示状态信息的位。其中的 4 个位 (C、V、Z 和 S) 可以用于条件跳转指令。2 个标志位 (H 和 D) 不能用于测试或者二进制转向十进制 (BCD) 的算法。余下的 2 个位用用户标志位 (F1 和 F2),可以设置位通用状态标志位。用户标志位对于算术操作是不受影响的,而且必须用指令设置或者清除。用户标志位不能用于条件跳转。在最初上电时没有被定义但是在复位时是不受影响的。图 1.1.2 表明了各个标志位在标志寄存器中的位置。

中断,不可屏蔽(陷阱)中断 (TRAP) 指令和非法指令将标志寄存器的值保存到堆栈。中断返回指令 (IRET) 将保存的值从堆栈返回到标志

图 1.1.2 标志寄存器

寄存器。

进位标志位:

算术运算结果产生进位或借位时,进位标志 C (Bit7) 置“1”,否则为“0”。一些循环或移位指令也影响进位标志寄存器。有 3 种指令可直接地变更进位标志寄存器的数值:进位位标志求补 (CCF)、重新设置进位位标志 (RCF)、设置进位位标志 (SCF)

零标志位:

算术和逻辑运算结果为零,此标志位 Z 置“1”,否则为“0”。在测试寄存器时,如果测试位是 00H,此标志位也置“1”,否则清零。如果循环和移位指令的执行结果为 00H,此标志位置“1”,否则清零。

符号标志位:

符号位 S 的值是执行完算术、逻辑、循环、移位指令后结果最高位的值。在执行带符号的算术运算指令时,用二进制 2 补码来表示和处理数据。符号位 (第 8 位) 为 0,则数据是正,因此符号标志也为 0。符号位为 1 时,则数据是负,符号标志为 1。

溢出标志位:

对于带符号位的算术、循环、移位运算,如果结果大于 127 或小于 -127,则溢出标志 V 为“1”。如没有溢出则此标志为“0”。执行逻辑指令操作溢出标志将清零。做完加法后,当操作数有相同的符号时,结果为相反的符号;做完减法后,如果两个操作数是有相反的符号,溢出标志是 1,而且结果的符号和源操作数的符号相同。做完循环操作后,如果目的操作数的符号位在旋转期间改变,溢出标志是 1。

十进制调整标志:

十进制调整标志 D 是用于 BCD 码算术运算的。由于加法和减法在 BCD 码运算上原理是不一样的,此标志表明了上一次执行的指令是一哪种,这样就保证以后的十进制调整运算正确。通常,此标志不能作测试条件码。做完减法后,此标志置“1”;做完加法后,此标志清零。

半进位标志:

当加法指令在第 4 位产生进位,或减法指令在第 4 位产生借位时,半进位标志 H 将置“1”。十进制调整指令 (DA) 将先前的加法或减法结果由二进制转为十进制时用到此标志。同十进制调整标志一样,用户不能直接地访问这个标志。

4. 条件码

C、Z、S 和 V 标志,控制条件跳转指令的操作。这 4 个标志可编成 4 位字段的 16 个常用的功能码,称为条件码 (cc),条件码形成条件指令的第 4 ~ 7 位。条件码及其所标志状态见表 1.1.2。一些二进制条件码可以用多余 1 个汇编代码。如果执行了条件跳转指令可以通过标志的结果测试操作的定义。

表 1.1.2 条件码

二进制	十六进制	助记符	定 义	标志测试操作
0000	0	F	不成立	-
0001	1	LT	小于	(S XOR V) = 1
0010	2	LE	小于或等于	(Z OR (S XOR V)) = 1
0011	3	ULE	无符号的小于或等于	(C OR Z) = 1
0100	4	OV	溢出	V = 1
0101	5	MI	负数	S = 1
0110	6	Z	零	Z = 1
0110	6	EQ	等于	Z = 1

续表

二进制	十六进制	助记符	定 义	标志测试操作
0111	7	C	进位	$C = 1$
0111	7	ULT	无符号的小于	$C = 1$
1000	8	T (or blank)	成立	-
1001	9	GE	大于或等于	$(S \text{ XOR } V) = 0$
1010	A	GT	大于	$(Z \text{ OR } (S \text{ XOR } V)) = 0$
1011	B	UGT	无符号的大于	$(C = 0 \text{ AND } Z = 0)$
1100	C	NOV	无溢出	$V = 0$
1101	D	PL	正数	$S = 0$
1110	E	NZ-Non Zero	不为零 $Z = 0$	
1110	E	NE	不相等	$Z = 0$
1111	F	NC	无进位	$C = 0$
1111	F	UGE	无符号的大于或等于	$C = 0$

5. 算术逻辑单元

算术逻辑单元 (ALU) 完成在数据上的算术和逻辑操作。算术操作包括加、减和乘。逻辑操作的功能包括二进制逻辑操作、位的移动和位的循环。

6. 位排序

对于多字节数据,eZ8 CPU 将高位字节存储在低位存储地址上。例如,数值 1 可以以 2 字节(16 位)数以 0001H 存储在寄存器对 123H 和 124H 上。其中数值的高位字节的 00H 存储在低位地址 123H 中,而数值的低位字节 01H 存储在高位地址 124H 中。这个多字节数据命令经常被认为是“最大的范围”。

第二章 eZ8 兼容性

一、概 述

eZ8 CPU 是美国 ZiLOG 公司所生产的流行的、容易使用的以及具有强大功能的 Z8 CPU 扩充和改良产品。已经使用过 Z8 CPU 的用户会很容易适应 eZ8 CPU。由于可以使用高级编程语言,比如说 C 语言,所有用户将会更加欣赏新的指令集,这些新的指令可以提高编程的效率。

二、汇编语言兼容性

eZ8 CPU 除了 WDH 指令外(看门狗定时器,当操作码为 4FH 时在 HALT 模式时的使能指令)均与 Z8 的汇编指令相同。已经拥有 Z8 汇编编码的用户可以很方便地使用他们已有的编码对 eZ8 CPU 进行操作。eZ8 的汇编器可以从 www.zilog.com 下载得到。

三、新 指 令

1. 新功能指令

表 1.2.1 列出了可以提供新功能的指令。

表 1.2.1 新功能指令

寄 存 器	指 令 描 述	寄 存 器	指 令 描 述
BCLR	位清除	CPC	带进位位的比较
BIT	位的设置或清除	LDC	传送常数
BRK	片上调试器暂停	LDCI	传送常数并自动增加地址
BSET	位设置	LEA	传送有效地址
BSWAP	位交换	MULT	8 位与 8 位相乘得出 16 位结果
BTJ	位的测试跳转	SRL	逻辑右移
BTJNZ	位测试非 0 时跳转	TRAP	软件陷阱
BTJZ	位测试 0 则跳转		

2. 扩展地址指令

新的扩展指令集允许在寄存器页之间进行数据传输。这个指令集允许生成 12 位的地址以及在 4KB 寄存器阵列地址空间中随机存储任意的寄存器值。表 1.2.2 列举出使用扩展寻址的指令集。

表 1.2.2 新增扩展地址指令

寄 存 器	指 令 描 述	寄 存 器	指 令 描 述
ADCX	使用扩展寻址带进位加	POPX	使用扩展寻址出栈
ADDX	使用扩展寻址加	PUSHX	使用扩展寻址进栈
ANDX	使用扩展寻址逻辑与	SBCX	使用扩展寻址的带进位减法
CPCX	带有扩展寻址的带进位的比较	SUBX	使用扩展方式的减法
CPX	使用扩展寻址比较	TCMX	使用扩展方式的逻辑“1”位测试
LDX	使用扩展寻址传送	TMX	使用扩展方式的逻辑“0”位测试
ORX	使用扩展寻址逻辑或	XORX	使用扩展寻址逻辑异或