

逻辑设计用 Z80 程序设计

(美) A. 奥斯本 J. 凯恩 著
R. 雷克托 S. 雅各布森

科学出版社

TP302
1026

逻辑设计用Z80程序设计

[美] A. 奥斯本 J. 凯 恩 著
R. 雷克托 S. 雅各布森

朱宗正 等 译

科学出版社

1986

内 容 简 介

本书介绍如何使用 Z80 汇编语言程序设计逻辑系统。主要内容包括：汇编语言与数字逻辑，数字逻辑的直接模拟，简单程序，从程序设计人员的角度看程序，Z80 指令系统，一些常用的子程序等。

书中列举了一个打字机逻辑系统的模拟设计，简要明确、形象具体、循序渐进、强调要领。对 Z80 指令系统的介绍完整细致，切合实用。

本书可供计算机专业人员、微型计算机使用部门、研究部门以及高等院校有关专业的师生参阅。

Adam Osborne Jerry Kane Russell Rector Susanna Jacobson
Z80 PROGRAMMING FOR LOGIC DESIGN

Osborne & Associates, Inc. 1978

逻辑设计用 Z80 程序设计

[美] A. 奥斯本 J. 凯 恩 著
R. 雷克托 S. 雅各布森

朱宗正 等译

责任编辑 孙月湘

科 学 出 版 社 出 版

北京朝阳门内大街 137 号

中国科学院印刷厂印刷

新华书店北京发行所发行 各地新华书店经售

*

1986年2月第 一 版

开本：787×1092 1/32

1986年2月第一次印刷

印张：14 7/8 插页：1

印数：0001—6,900

字数：334,000

统一书号：15031·695

本社书号：4471·15-3

定价：3.50 元

逻辑设计

译者序

微型计算机以其灵巧、方便、易学、适用、可靠、低价而广受欢迎。微处理器和微型计算机自七十年代初问世以来，其应用已迅速遍及军事、科技、生产、管理、通信、控制、服务、咨询、医疗、诊断、教学、游戏直至文艺创作、家庭生活等各个方面。

为了提高工作效率、经济效果和社会效益，国内外正掀起一场“微型计算机热”。这也正是我国引入并利用新技术来加速实现四化的良好时机。

大量微型计算机的使用不但需要有成千上万从事计算机及其应用的专业人员，同时还需有为数更多、领域广泛的使用微型计算机的非专业人员。对于后者，极大部分都属于新手；对于前者，也急需扩充队伍。为了适应形势的需要，我们翻译了本书。

这本书是美国奥斯本（Osborne）公司出版的“微型计算机丛书”之一。书中论述了用 Z 80 微型计算机的汇编语言程序模拟典型的逻辑器件和一个实用逻辑系统的功能及工作过程。取材实用、主次分明、抓住典型、讲解清楚。突出了设计中的要领，对初学者十分有益；书中还细述了每条指令的用法，对专业人员亦有参考价值。

Z 80 是国际市场上 8 位微处理器的一个主要品种，国内也已有大量用户和以它为 CPU 的各种微型计算机产品。它丰富灵活的指令系统适合对逻辑系统的模拟。用 Z 80 汇编语言编写成模拟程序，比用高级语言编写的程序具有空间紧凑、

运行快速、关系简单、修改容易等优点。因此，熟悉 Z 80 汇编语言程序设计对逻辑设计的模拟将十分有利于逻辑设计方面的工作开展。同时也为程序设计人员展示了新的工作方面。

曾经有许多单位和个人向我们索取 Z 80 汇编语言的程序设计教材和资料，遗憾的是当时未能满足。现在谨向各位推荐此中译本供作参考。

参加本书译校工作的有朱宗正、周克勤、曹长春、刘文英和徐加祁等同志。译者对原文中的个别差错作了校正；出于中文编排上的考虑，在结构和形式上作了微小调整。但因水平有限，加之整个译校工作都是业余断续地进行的，其中难免有疏漏不妥之处，恳请读者指正。

译者

1984年9月

目 录

第一章 绪言	1
1.1 阅读本书的预备知识	1
1.2 懂得汇编语言	2
第二章 汇编语言与数字逻辑	3
2.1 设计周期	3
2.2 数字逻辑的模拟	7
2.3 信号反相器的微型计算机模拟	8
2.3.1 微型计算机中的事件序列	8
2.3.2 实现变换作用	10
2.3.3 确定数据源和目标	11
2.3.4 事件的定时	17
2.4 缓冲器、放大器和信号的负载	20
2.5 7404/05/06六反相器的微型计算机模拟	29
2.6 7408/09四双输入正与门的微型计算机模拟	31
2.6.1 双输入的功能	32
2.7 7411三三输入正与门的微型计算机模拟	34
2.7.1 三输入的功能	35
2.7.2 尽量减少CPU寄存器的存取次数	38
2.7.3 存储器的利用和执行速度的比较	43
2.8 7474双D型可预置和清除的正沿触发触发器的 微型计算机模拟	45
2.8.1 触发器的数字逻辑说明	46
2.8.2 触发器的汇语言模拟	50
2.8.3 一般触发器微型计算机模拟	51
2.9 实时器件的微型计算机模拟	52
2.9.1 555单稳多谐振荡器	53

2.9.2	74121单稳多谐振荡器	54
2.9.3	74107双装带清除的J-K主从触发器	58
2.9.4	实际时间的微型计算机模拟	60
2.9.5	微型计算机定时用的指令循环	61
2.9.6	数字逻辑模拟的范围	67
2.9.7	同外部单脉冲发生器的连接	67
2.9.8	时间输出和中断	70
2.9.9	同可编程序的时间的连接	70

第三章 直接的数字逻辑模拟72

3.1 奎米打印机是怎样工作的73

3.2 输入和输出信号80

3.2.1 输入/输出器件81

3.2.2 Z 80并行 I/O接口 PIO81

3.2.3 输入信号89

3.2.4 输入信号小结100

3.2.5 输出信号100

3.3 面向数字逻辑的模拟102

3.3.1 逻辑概观102

3.3.2 FFA_w 触发器104

3.3.3 触发器FFA_w的模拟106

3.3.4 触发器FFB_w118

3.3.5 触发器FFB的模拟121

3.3.6 触发器FFC129

3.3.7 触发器FFC的模拟131

3.3.8 START RIBBON MOTION (色带启动) 脉冲的模拟135

3.3.9 触发器FFD137

3.3.10 触发器FFD的模拟137

3.3.11 触发器FFE141

3.3.12 PW SETTLING (打印字轮定位)的单脉冲发生器144

3.3.13 PW SETTLING 单脉冲发生器的模拟145

3.3.14 触发器FFF146

3.3.15 触发器FFF的模拟148

3.3.16 555多谐振荡器152

3.3.17	多谐振荡器555的模拟	153
3.3.18	PW RELEASE ENABLE(字轮解脱可用)触发器	163
3.3.19	PW RELEASE ENABLE(字轮解脱可用)触发器的 模拟	164
3.3.20	PW READY ENABLE 单脉冲发生器	166
3.4	关于模拟的总结	170
第四章	简单程序	176
4.1	从不同的角度看问题	176
4.1.1	汇编语言的定时与数字逻辑的定时	176
4.1.2	输入和输出信号	177
4.2	微型计算机的器件结构	180
4.2.1	通常的设计概念	180
4.2.2	Z80并行输入/输出接口 (PIO)	182
4.2.3	ROM和RAM存储器	184
4.2.4	系统的初始化	186
4.2.5	程序流程图	186
4.2.6	程序的逻辑错误	208
4.2.7	复位和初始化	213
4.3	程序的综合	215
第五章	程序设计者的看法	220
5.1	简单程序设计的效率	220
5.1.1	高效率查表	220
5.2	子程序	227
5.2.1	子程序的调用	228
5.2.2	子程序的返回	234
5.2.3	如果使用子程序	235
5.2.4	子程序的有条件返回	237
5.2.5	子程序的多路返回	240
5.2.6	子程序的有条件调用	245
5.3	宏	247
5.3.1	什么叫宏?	248
5.3.2	带参量的宏	250
5.4	中断	251

5.4.1 中断的硬件考虑	252
5.4.2 多级中断	266
5.4.3 中断的合理性	269

第六章 Z80指令系统271

6.1 基本规定和说明271

6.1.1 缩写符号271

6.1.2 状态274

6.1.3 指令记忆符275

6.1.4 指令目标码275

6.1.5 指令的执行时间和代码275

6.2 指令细述306

6.2.1 ADC A, data——累加器加立即数及进位306

6.2.2 ADC A, reg——累加器加寄存器及进位307

6.2.3 ADC A, (HL)——累加器加存储器及进位

ADC A, (IX + disp)308

ADC A, (IY + disp)308

6.2.4 ADC HL, rp——H和L加寄存器对及进位309

6.2.5 ADD A, data——累加器加立即数310

6.2.6 ADD A, reg——累加器加寄存器311

6.2.7 ADD A, (HL)——累加器加存储器

ADD A, (IX + disp)313

ADD A, (IY + disp)313

6.2.8 ADD HL, rp——H和L加寄存器对314

6.2.9 ADD xy, rp——变址寄存器加寄存器对315

6.2.10 AND data——累加器同立即数逻辑与316

6.2.11 AND reg——累加器同寄存器逻辑与317

6.2.12 AND (HL)——累加器同存储器逻辑与

AND (IX + disp)318

AND (IY + disp)318

6.2.13 BIT b, reg——检测寄存器reg中的二进制位b319

6.2.14 BIT b, (HL)——检测所指存储单元中的二进制位b

BIT b, (IX + disp)320

BIT b, (IY + disp)320

6.2.15	CALL label——调用由操作码指明的子程序	322
6.2.16	CALL condition, label——若条件满足, 调用由操作码指明的子程序	323
6.2.17	CCF——进位标志取反	324
6.2.18	CP data——累加器的内容与立即数比较	324
6.2.19	CP reg——累加器与寄存器比较	325
6.2.20	CP (HL)——累加器与存储器比较	
	CP (IX + disp)	
	CP (IY + disp)	326
6.2.21	CPD——累加器与存储器比较。地址和字节计数器各减1	328
6.2.22	CPDR——累加器与存储器比较。地址和字节计数器各减1。连续进行到条件相符或字节计数器为0	329
6.2.23	CPI——累加器与存储器比较。字节计数器减1。地址加1	330
6.2.24	CPIR——累加器与寄存器比较。字节计数器减1。地址加1。连续进行到条件相符或字节计数器为0	331
6.2.25	CPL——累加器取反	332
6.2.26	DAA——累加器的十进制校正	332
6.2.27	DEC reg——寄存器的内容减1	333
6.2.28	DEC rp——所指寄存器对的内容减1	
	DEC IX	
	DEC IY	334
6.2.29	DEC (HL)——存储单元的内容减1	
	DEC (IX + disp)	
	DEC (IY + disp)	336
6.2.30	DI——封锁中断	337
6.2.31	DJNZ disp——若寄存器B的内容非0, 则相对于程序计数器的当前内容进行跳转	338
6.2.32	EI——开放中断	338
6.2.33	EX AF, AF'——程序的状态和替换程序的状态交换	340
6.2.34	EX DE, HL——DE和HL的内容交换	341
6.2.35	EX (SP), HL——栈顶和寄存器的内容交换	
	EX (SP), IX	
	EX (SP), IY	342
6.2.36	EXX——寄存器对和替换寄存器对交换	343

6.2.37	HALT——暂停	344
6.2.38	IM 0——0 式中断	344
6.2.39	IM 1——1 式中断	345
6.2.40	IM 2——2 式中断	345
6.2.41	IN A, (port)——输入到累加器	345
6.2.42	INC reg——寄存器的内容加 1	346
6.2.43	INC rp——所指寄存器的内容加 1	346
	INC IX	
	INC IY	347
6.2.44	INC (HL)——存储器的内容加 1	
	INC (IX + disp)	
	INC (IY + disp)	348
6.2.45	IND——输入到存储器, 指针减 1	349
6.2.46	INDR——输入到存储器, 指针减 1, 连续进行到字节 计数器为 0	350
6.2.47	INI——输入到存储器, 指针加 1	351
6.2.48	INIR——输入到存储器, 指针加 1. 连续进行到字节 计数器为 0	351
6.2.49	IN reg, (C)——输入到寄存器	352
6.2.50	JP label——跳转到由操作码标识的指令	353
6.2.51	JP condition, label——若条件满足, 则跳转到由操作 码标识的地址	354
6.2.52	JP (HL)——跳转到由 16 位寄存器内容指明的地址 JP (IX)	
	JP (IY)	355
6.2.53	JR C, disp——若进位为 1, 相对于程序计数器的 内容进行跳转	356
6.2.54	JR disp——相对于程序计数器的内容进行跳转	357
6.2.55	JR NC, disp——若进位标志为 0, 相对于程序计数器的 内容进行跳转	358
6.2.56	JR NZ, disp——若零标志是 0, 相对于程序计数器的 内容进行跳转	358
6.2.57	JR Z, disp——若零标志是 1, 相对于程序计数器的 内容进行跳转	359
6.2.58	LD A, IV——中断向量或更新寄存器的内容移入 累加器	
	LD A, R	359

6.2.59	LD A, (addr)——用直接寻址从存储单元输入累加器	360
6.2.60	LD A, (rp)——由按寄存器对内容编址的存储单元 输入累加器	361
6.2.61	LD dst, src——源寄存器的内容移入目标寄存器	362
6.2.62	LD HL, (addr)——用直接寻址从存储单元输入到寄 存器对或变址寄存器	
	LD rp, (addr)	
	LD IX, (addr)	
	LD IY, (addr)	363
6.2.63	LD IV, A——从累加器输入到中断向量或更新 寄存器	
	LD R, A	365
6.2.64	LD reg, data——立即数输入寄存器	366
6.2.65	LD rp, data——16位的立即数输入寄存器	
	LD IX, data	
	LD IY, data	367
6.2.66	LD reg, (HL)——从存储器输入寄存器	
	LD reg, (IX + disp)	
	LD reg, (IY + disp)	368
6.2.67	LD SP, HL——HL或变址寄存器的内容移入堆栈 指示器(指针)	
	LD SP, IX	
	LD SP, IY	370
6.2.68	LD (addr), A——累加器的内容用直接寻址存入 存储器	371
6.2.69	LD (addr), HL——寄存器对或变址寄存器的内容 用直接寻址存入存储器	
	LD (addr), rp	
	LD (addr), xy	372
6.2.70	LD (HL), data——立即数输入存储器	
	LD (IX + disp)	
	LD (IY + disp)	374
6.2.71	LD (HL), reg——从寄存器输入存储器	
	LD (IX + disp), reg	
	LD (IY + disp), reg	375
6.2.72	LD (rp), A——从累加器输入到由寄存器对编址的 存储单元	376

6.2.73	LDD——存储单元之间传送数据。目标和源地址都减1	377
6.2.74	LDDR——存储单元之间传送数据,连续进行到字节计数器为0。目标和源地址都减1	378
6.2.75	LDI——存储单元之间传送数据。目标和源地址都加1	379
6.2.76	LDIR——存储单元之间传送数据,直至字节计数器为0。目标和源地址都加1	380
6.2.77	NEG——累加器的内容取补	381
6.2.78	NOP——空操作	381
6.2.79	OR data——累加器同立即数逻辑或	382
6.2.80	OR reg——累加器同寄存器逻辑或	383
6.2.81	OR (HL)——累加器同存储器逻辑或 OR (IX + disp) OR (IY + disp)	384
6.2.82	OUT (C), reg——从寄存器输出	385
6.2.83	OUTD——从存储器输出,地址减1	386
6.2.84	OTDR——从存储器输出,地址减1。连续进行到寄存器B = 0	387
6.2.85	OUTI——从存储器输出,地址加1	388
6.2.86	OTIR——从存储器输出,地址加1。连续进行到寄存器B = 0	388
6.2.87	OUT (port), A——从累加器输出	389
6.2.88	POP rp——从栈顶读出 POP IX POP IY	390
6.2.89	PUSH rp——向栈顶写入 PUSH IX PUSH IY	391
6.2.90	RES b, reg——存储单元的指定位置0	393
6.2.91	RES b, (HL)——存储单元的指定位置0 RES b, (IX + disp) RES b, (IY + disp)	394
6.2.92	RET——从子程序返回	395
6.2.93	RET cond——若条件满足,从子程序返回	396
6.2.94	RETI——从中断返回	397
6.2.95	RETN——从不可屏蔽的中断返回	398
6.2.96	RL reg——寄存器的内容连同进位循环左移	398
6.2.97	RL (HL)——存储单元的内容连同进位循环左移	

RL (IX + disp)	
RL (IY + disp)	399
6.2.98 RLA——累加器的内容连同进位循环左移	401
6.2.99 RLC reg——寄存器的内容左送进位位且循环左移	401
6.2.100 RLC (HL)——存储单元的内容左送进位位且 循环左移	
RLC (IX + disp)	
RLC (IY + disp)	402
6.2.101 RLCA——累加器的内容左送进位位且循环左移	404
6.2.102 RLD——累加器和存储单元之间循环左移二—十 进制数一位	405
6.2.103 RR reg——寄存器的内容连同进位循环右移	406
6.2.104 RR (HL)——存储单元的内容连同进位循环右移	
RR (IX + disp)	
RR (IY + disp)	407
6.2.105 RRA——累加器的内容连同进位循环右移	408
6.2.106 RRC reg——寄存器的内容右送进位位且循环右移	409
6.2.107 RRC (HL)——存储单元的内容右送进位位且 循环右移	
RRC (IX + disp)	
RRC (IY + disp)	410
6.2.108 RRC A——累加器的内容右送进位位且循环右移	411
6.2.109 RRD——累加器和存储单元之间循环右移二—十 进制数一位	412
6.2.110 RST——重新启动	413
6.2.111 SBC A, data——累加器减立即数及借位	414
6.2.112 SBC A, reg——累加器减寄存器及借位	415
6.2.113 SBC A, (HL)——累加器减存储单元及借位	
SBC A, (IX + disp)	
SBC A, (IY + disp)	416
6.2.114 SBC HL, rp——H和L减寄存器对的内容及借位	418
6.2.115 SCF——进位标志置1	419
6.2.116 SET b, reg——所指的寄存器某位置1	419
6.2.117 SET b, (HL)——所指的存储单元某位置1	
SET b, (IX + disp)	
SET b, (IY + disp)	420
6.2.118 SLA reg——寄存器内容算术左移	422

6.2.119	SLA (HL)——存储单元内容算术左移	
	SLA (IX + disp)	
	SLA (IY + disp)	423
6.2.120	SRA reg——寄存器内容算术右移	424
6.2.121	SRA (HL)——存储单元内容算术右移	
	SRA (IX + disp)	
	SRA (IY + disp)	425
6.2.122	SRL reg——寄存器内容逻辑右移	427
6.2.123	SRL (HL)——存储单元内容逻辑右移	
	SRL (IX + disp)	
	SRL (IY + disp)	428
6.2.124	SUB data——累加器减立即数	429
6.2.125	SUB reg——累加器减寄存器	430
6.2.126	SUB (HL)——累加器减存储单元	
	SUB (IX + disp)	
	SUB (IY + disp)	431
6.2.127	XOR data——累加器同立即数异或	432
6.2.128	XOR reg——累加器同寄存器异或	433
6.2.129	XOR (HL)——累加器与存储器异或	
	XOR (IX + disp)	
	XOR (IY + disp)	434

第七章 公用子程序 436

7.1 存储器寻址 436

7.1.1 间接寻址 437

7.1.2 间接后变址的寻址 438

7.2 数据传送 438

7.2.1 传送简单的数据块 438

7.2.2 多重表的查找 440

7.2.3 数据分类 442

7.3 算术 444

7.3.1 二进制加法 444

7.3.2 二进制减法 448

7.3.3 十进制加法 448

7.3.4 十进制减法 449

7.4 乘法和除法 449

7.4.1	8 位二进制乘法	449
7.4.2	一个 8 位的二进制乘法程序	453
7.4.3	16 位的二进制乘法	455
7.4.4	二进制除法	456
7.5	程序执行的顺序逻辑	458
7.5.1	转移表	458

第一章 绪 言

本书讲解汇编语言程序在微型计算机系统中如何能代替组合逻辑。组合逻辑由流行的不能编程的逻辑器件，如标准7400系列数字逻辑组成。

倘若你从事于逻辑设计，本书将教给你如何用新的方法——在微型计算机系统中编出汇编程序，去完成你原先的工作。

倘若你致力于程序设计，本书将告诉你程序设计如何找到新的用途——在逻辑设计中的应用。

这是一本“如何使用”的书，因此，内容写得十分具体详细，而且还直接引用一台实际的微型计算机——Z80。

1.1 阅读本书的预备知识

本书是《微型计算机入门 (An Introduction to Microcomputers)》的续编。《微型计算机入门》的初版是一册，第二版有两册。

《微型计算机入门》概念性地介绍微处理器和微型计算机，不讲概念实现的实际问题。而本书谈及实施中的实际问题。对于本续编，仅作一个假定，假定读者已经读过，或从别处知道整个《微型计算机入门》的前后内容。然而，设计者在开始进入实际设计之前，必须拥有厂商所提供的资料，资料中详载选用的器件。

注意，实际上硬件和定时在本书中都没有讲，也未讲