

8080 程序设计在逻辑 设计中的应用

〔美〕 亚当·奥斯本 著

赵 辰 等译

人 民 邮 电 出 版 社

8080 PROGRAMMING FOR LOGIC DESIGN

by Adam Osborne

1976

内 容 提 要

本书是《微型计算机入门》丛书的续篇，主要内容是讲述利用汇编语言程序来代替组合逻辑的程序设计。内容叙述具体、实用。读者可通过举出的典型例子，领会编制汇编语言程序的要点，熟悉常用典型程序的用法。书中还结合实际讲解了子程序、宏程序和中断程序的编制，详细地介绍了 Intel 8080 每条指令的功能和用法。同时还介绍了一些常用的子程序。这些对于初次接触微型计算机应用和汇编语言程序设计的读者来说，都有实用参考价值。本书可供学习程序编制的科技人员、大专院校师生及初学微型计算机应用的工作人员阅读。

本书由赵辰同志担任全书的译校工作，参加翻译的有王钟馨、幸云辉、陈郁青、谢楚屏、李怀诚、孙德生、高锡武等同志。

8080程序设计在逻辑设计中的应用

〔美〕亚当·奥斯本 著

赵 辰 等译

人民邮电出版社出版

北京东长安街27号

北京印刷一厂印刷

新华书店北京发行所发行

各地新华书店经售

*

开本：787×1092 1/32 1983年9月第一版
印张：13 24/32 页数：220 1983年9月北京第一次印刷
字数：310千字 插页：1 印数：1—18,000册

统一书号：15045·总2704-有5288

定价：1.70元

目 录

第一章 引言	1
1-1 这本书假定你已经知道了...	2
1-2 了解汇编语言	2
1-3 这本书是怎样排印的	3
第二章 汇编语言和数字逻辑	4
2-1 设计周期	4
2-2 模拟数字逻辑	8
2-3 用微型计算机模拟信号反相器	9
2-3-1 微型计算机事件序列	9
2-3-2 传递函数的实现	11
2-3-3 确定数据的源和目的地	11
2-3-4 事件的定时	18
2-4 缓冲器、放大器和信号负载	21
2-5 用微型计算机模拟 7404/05/06/07 六反相器	30
2-6 用微型计算机模拟 7408/09 二输入、 四正“与”门	31
2-6-1 二输入功能	32
2-7 用微型计算机模拟 7411 三输入、三正“与”门	34
2-7-1 三输入功能	35
2-7-2 尽可能减少对 CPU 寄存器的存取	38
2-7-3 存储器利用率和执行速度的比较	41
2-8 用微型计算机模拟可预置和可清除的 7474 正跳沿触发双 D 触发器	42
2-8-1 触发器的数字逻辑描述	42

2-8-2	用汇编语言对触发器的一种模拟	45
2-8-3	用微型计算机模拟一般触发器	47
2-9	用微型计算机模拟实时器件	47
2-9-1	555 单稳多谐振荡器	48
2-9-2	74121 单稳多谐振荡器	49
2-9-3	74107 带清零端的双 J-K 主从触发器	52
2-9-4	用微型计算机模拟实时	54
2-9-5	微型计算机定时指令循环	54
2-9-6	对于数字逻辑模拟的局限性	60
2-9-7	与外部单冲触发电路的接口	60
2-9-8	逾时和中断	63

第三章 数字逻辑的直接模拟 64

3-1	QUME 打印机是如何工作的	65
3-2	输入/输出信号	71
3-2-1	输入/输出器件	71
3-2-2	8255 可编程序外部接口	72
3-2-3	8212 八位输入/输出口	76
3-2-4	输入信号	78
3-2-5	返回选通 (RETURN STROBE)	78
3-2-6	阻止“打印锤启动”释放 (PFL REL)	80
3-2-7	色带准备提升 (RIB LIFT RDY)	80
3-2-8	印字轮选通 (PW STROBE)	81
3-2-9	“启动色带移动脉冲” (FFA)	82
3-2-10	复位 (RESET)	83
3-2-11	输纸轴释放 (PFR REL)	84
3-2-12	输纸托架释放 (CA REL)	84
3-2-13	FFI	85
3-2-14	色带用完 (EOR DET)	86
3-2-15	允许打印锤触发器 (HAMMER ENABLE FF)	89

3-2-16	时钟	89
3-2-17	H1—H6	90
3-2-18	输入信号小结	90
3-2-19	输出信号	91
3-3	面向数字逻辑电路的模拟	92
3-3-1	对于逻辑的简要说明	92
3-3-2	触发器 FFA _w	94
3-3-3	触发器 FFA _w 的模拟	97
3-3-4	触发器 FFB _w	107
3-3-5	触发器 FFB 的模拟	110
3-3-6	触发器 FFC	121
3-3-7	触发器 FFC 的模拟	124
3-3-8	“启动色带移动”脉冲的模拟	128
3-3-9	触发器 FFD	132
3-3-10	触发器 FFD 的模拟	132
3-3-11	触发器 FFE	137
3-3-12	“稳定印字轮”单冲触发器	141
3-3-13	“稳定印字轮”单冲触发器的模拟	142
3-3-14	触发器 FFF	144
3-3-15	触发器 FFF 的模拟	146
3-3-16	555多谐振荡器	151
3-3-17	555多谐振荡器的模拟	152
3-3-18	“允许印字轮释放”触发器	165
3-3-19	“允许印字轮释放”触发器的模拟	165
3-3-20	“允许印字轮准备”单冲触发器的模拟	169
3-4	关于模拟的小结	175
第四章	一个简单程序	185
4-1	汇编语言与数字逻辑在定时关系上的比较	185
4-2	输入和输出信号	186

4-3 微型计算机器件的组态	188
4-3-1 一般的设计概念	189
4-3-2 8255 可编程序外部接口	190
4-3-3 系统初始化	193
4-3-4 只读存储器和随机存取存储器	193
4-3-5 程序流程图	195
4-3-6 程序逻辑错误	219
4-3-7 复位和初始化	226
4-4 程序小结	227
第五章 程序人员的看法	234
5-1 简单程序设计的效率	234
5-1-1 高效率的查表方法	234
5-2 硬件的利用	241
5-2-1 硬件专用指令	242
5-2-2 硬件特性的直接用途	244
5-3 子程序	246
5-3-1 子程序调用	249
5-3-2 子程序的返回	252
5-3-3 什么时候使用子程序	256
5-3-4 子程序的条件返回	257
5-3-5 多重子程序返回	262
5-3-6 子程序的条件调用	268
5-4 宏指令	270
5-4-1 什么是宏指令?	271
5-4-2 具有参数的宏指令	272
5-5 中断	275
5-5-1 中断硬件的考虑	276
5-5-2 中断服务程序	279
5-5-3 对于中断的评价	287

5-5-4 多级中断	289
第六章 8080/9080 指令系统	292
6-1 缩写符号	292
6-2 状态	303
6-3 指令的结果代码	305
6-4 指令的执行时间和代码	305
表 6-1 8080/9080 微型计算机指令系统摘要	293
表 6-2 指令结果代码和执行周期摘要	304
6-5 ACI——立即数和累加器进行带进位的加法	305
6-6 ADC——寄存器或存储器和累加器进行带进位的加法	306
6-7 ADD——寄存器或存储器和累加器相加	308
6-8 ADI——立即数和累加器相加	309
6-9 ANA——寄存器或存储器和累加器相“与”	310
6-10 ANI——立即数和累加器相“与”	312
6-11 CALL——调用由操作数标示的子程序	313
6-12 CC——调用由操作数标示的子程序，但仅当进位位状态等于 1 时才调用	314
6-13 CM——调用由操作数标示的子程序，但仅当符号状态位等于 1 时才调用	315
6-14 CMA——累加器内容取反	315
6-15 CMC——进位位状态取反	316
6-16 CMP——寄存器或存储器和累加器相比较	317
6-17 CNC——调用由操作数标示的子程序，但仅当进位位状态等于 0 时才调用	319
6-18 CNZ——调用由操作数标示的子程序，但仅当零状态等于 0 时才调用	320

6-19	CP——调用由操作数标示的子程序，但仅 当符号位状态等于 0 时才调用	320
6-20	CPE——调用由操作数标示的子程序，但仅 当奇偶位状态等于 1 时才调用	321
6-21	CPI——立即数与累加器内容相比较	322
6-22	CPO——调用由操作数标示的子程序，但仅 当奇偶位状态等于 0 时才调用	323
6-23	CZ——调用由操作数标示的子程序，但仅 当零状态位等于 1 时才调用	323
6-24	DAA——十进制调整累加器	324
6-25	DAD——寄存器对和 H,L 相加	325
6-26	DCR——寄存器或存储器的内容减 1	326
6-27	DCX——寄存器对减 1	328
6-28	DI——禁止中断(关中断)	330
6-29	EI——允许中断(开中断)	330
6-30	HLT——暂停	333
6-31	IN——输入累加器	334
6-32	INR——寄存器或存储器内容增 1	334
6-33	INX——寄存器对增 1	336
6-34	JC——若有进位，转移	338
6-35	JM——若为负，转移	338
6-36	JMP——转移到由操作数标示的指令	339
6-37	JNC——若无进位，转移	340
6-38	JNZ——若不为零，转移	340
6-39	JP——若为正，转移	341
6-40	JPE——若奇偶位状态为偶，转移	341
6-41	JPO——若奇偶位状态为奇，转移	342

6-42	JZ——若结果为零, 转移	342
6-43	LDA——将直接寻址的存储单元内容送累加器 ..	343
6-44	LDAX——由寄存器对寻址的存储单元内容 送累加器	344
6-45	LHLD——将直接寻址的存储单元内容送入 寄存器 H 和 L	345
6-46	LXI——将 16 位立即数送入寄存器对	347
6-47	MOV——传送数据	347
6-48	MVI——将立即数据送入寄存器或存储器	350
6-49	NOP——不操作	352
6-50	ORA——寄存器或存储器和累加器相“或”	353
6-51	ORI——立即数和累加器内容相“或”	355
6-52	OUT——从累加器输出	356
6-53	PCHL——转移到由寄存器 HL 所指定的地址	357
6-54	POP——从栈顶读出	358
6-55	PUSH——写入栈顶	359
6-56	RAL——累加器内容连同进位位循环左移	360
6-57	RAR——累加器内容连同进位位循环右移	362
6-58	RC——若进位位状态等于 1, 返回	363
6-59	RET——从子程序返回	364
6-60	RLC——累加器循环左移	365
6-61	RM——若符号位状态等于 1, 返回	366
6-62	RNC——若进位位状态等于 0, 返回	366
6-63	RNZ——若零状态位等于 0, 返回	367
6-64	RP——若符号状态位等于 0, 返回	368
6-65	RPE——若奇偶状态位等于 1, 返回	369

6-66	RPO——若奇偶状态位等于 0, 返回	369
6-67	RRC——累加器循环右移	370
6-68	RST——重新启动	371
6-69	RZ——若零状态位等于 1, 返回	372
6-70	SBB——累加器内容与寄存器或存储器内容 进行带借位减法	373
6-71	SBI——累加器与立即数据进行带借位减法	375
6-72	SHLD——直接存入 H 和 L 寄存器	376
6-73	SPHL——将寄存器 H 和 L 的内容送入栈指示器	377
6-74	STA——累加器内容存入采用直接寻址的 存储单元	377
6-75	STAX——累加器内容存入由寄存器对指定 的存储单元	379
6-76	STC——置进位位状态	380
6-77	SUB——从累加器中减去寄存器或存储器 的内容	380
6-78	SUI——从累加器减去立即数据	382
6-79	XCHG——交换寄存器 DE 和 HL 的内容	383
6-80	XRA——寄存器或存储器和累加器相“异”	384
6-81	XRI——立即数据和累加器相“异”	386
6-82	XTHL——栈顶和 H, L 交换	387
第七章 一些常用的子程序		389
7-1	存储器寻址	389
7-1-1	自动增 1 和自动减 1	390
7-1-2	变址寻址	393
7-1-3	间接寻址	394
7-1-4	变址后间接寻址	395

7-2 数据传送	395
7-2-1 传送简单的数据块	395
7-2-2 多重查表	397
7-2-3 数据分类	399
7-3 运算	402
7-3-1 二进制加法	403
7-3-2 二进制减法	406
7-3-3 十进制加法	406
7-3-4 十进制减法	407
7-4 乘法和除法	409
7-4-1 八位二进制乘法	409
7-4-2 八位二进制除法	413
7-4-3 十六位二进制乘法	415
7-4-4 二进制除法	416
7-5 程序执行顺序逻辑转移表	418
附录	
A 标准字符代码	421

第一章 引言

组合逻辑

本书解释在微型计算机系统中如何利用汇编语言程序来代替组合逻辑，也就是如何代替象 7400 系列标准数字逻辑集成电路那样现成的、非可编程序的逻辑器件所组成的电路。

假如你是一位逻辑设计者，本书将教给你如何使用新的方法，用微型计算机系统中产生的汇编语言程序来从事你的旧业。

假如你是一位程序设计者，本书将告诉你程序设计已经找到一种新的用途：用于逻辑设计。

这是一本教你如何实际动手的书。这样，它的内容也就必须十分具体，所以直接以 8080 A 这种类型的微型计算机作为具体对象来加以讨论。生产 8080 A 型微型计算机的厂商有好几家，它们的产品有：

AMD 9080 A

INTEL 8080 A

NEC 8080 A

TMS 8080 A

NS 8080 A

生产这些微型计算机的厂商是：

INTEL 有限公司

高级微型器件公司

德克萨斯仪器公司

· NEC 微型计算机公司

国家半导体公司

1111229

• 1 •

1-1 这本书假定你已经知道了...

本书是“微型计算机入门”的续篇。“入门”初版时为一册，再版时分为两册。

“微型计算机入门”一书介绍微处理机和微型计算机的概念；它并不涉及实现这些概念的实际手段。而本书则介绍了实现这些概念的实际手段。

由于本书是“入门”的续篇，所以就简单地假定你已阅读过或已了解“微型计算机入门”一书所包括的内容。不过，在你着手设计一个实际的方案之前，你还需要阅读具体介绍你所选用的各种器件的说明资料。

尤其应当注意的是，这本书并不介绍 8080 A/9080 A 中央处理机，或任何其它微型计算机器件的硬件和定时关系。由于本书的整个目的是介绍程序设计，所以第六章介绍 8080 A/9080 A 指令系统。

1-2 了解汇编语言

汇编语言指令是微型计算机系统的传递函数，把它们组合起来，就构成了一套“指令系统”，用来描述微型计算机所能完成的各种操作。

你把应该在微型计算机里依次发生的事件定义为一个指令序列。那末，把这些指令序列组合起来就构成了一个汇编语言程序。

实际上，理解各条指令在微型计算机里做些什么是十分简单的；它是应用微型计算机的最简单的一个方面。然而，它会使初次编制程序的用户不知所措。如果其中也包括你自己，那末我们要奉劝你，如果忘记了助记符和指令系统也不要紧，当

你在本书中遇到它们时，一条一条地去查阅指令就行。如果你不了解这条指令是做什么的话，就请参阅一下第六章。

只有当你惧怕程序设计时，它的幽灵才会纠缠着你。

1-3 这本书是怎样排印的

请注意，本书正文是用粗体字和细体字印刷的。这是为了帮助你跳过本书中你所熟悉的那部分内容。细体字所阐述的，只是前面粗体字所介绍知识的延伸。因而，最初只需阅读粗体字，直到你需要知道得更多时，你再开始阅读细体字的内容。

本书中译本分别用老宋体(对应于原书的粗体字)和仿宋体(对应于原书的细体字)印刷，这样在阅读时，亦可根据需要加以区别。

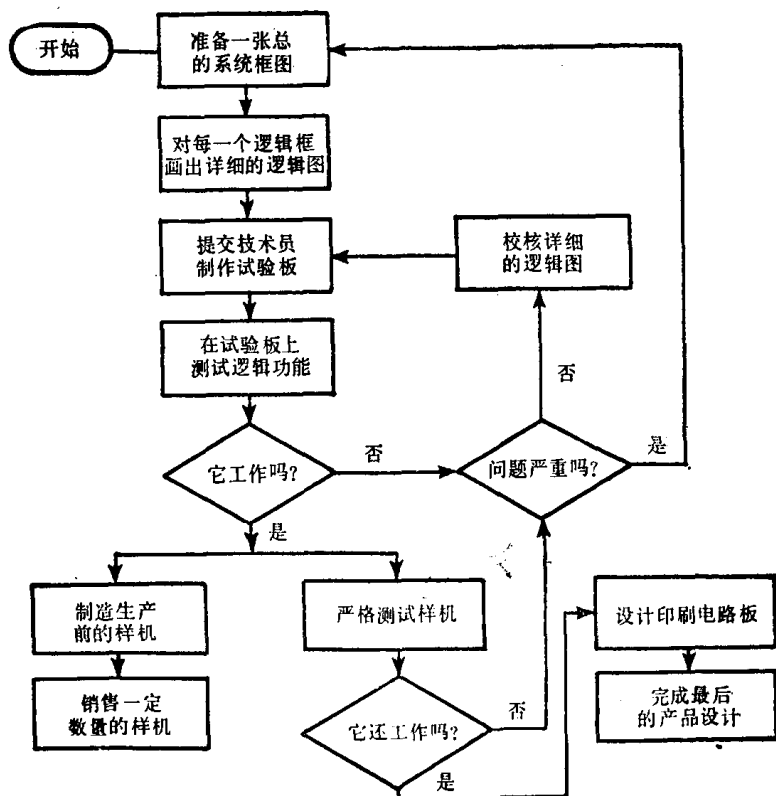
第二章 汇编语言和数字逻辑

2-1 设计周期

数字逻辑 设计周期

任何由分立的数字逻辑元件组成的产品都要经过一个妥善规定好的设计周期。

让我们假定已经根据市场销售的观点制定



出产品规格。

已经取得一份列举必要的产品性能和特性的产品说明书。
你的任务是提供一套能适合生产的设计。

设计周期如第4页的流程图所示。

在任何数字逻辑设计周期中都有一个代价昂贵而又缓慢的重复循环；如上图所示。它包括下列步骤：

重新绘制逻辑图

重新制做一块新的试验板

检测试验板是否有逻辑错误、技术性错误或组件缺陷。

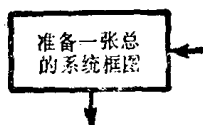
这些重复循环使得组合逻辑设计缓慢而昂贵。这不仅在设计的初始阶段如此，而且当你以后决定修改或提高产品质量时更会如此。

微型计算机 逻辑设计周期

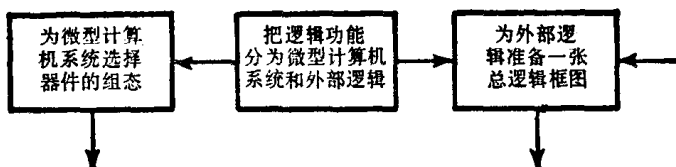
当你开始使用微型计算机时会发生什么情况呢？首先，你的一部分逻辑功能消失在一个“黑盒子”当中。它就是微型计算机系统：



你所要进行的第一步：



现在必须把它再做如下的划分：



如果你不了解微型计算机能做什么的话，那么要把你的应用课题划成微型计算机系统和外部数字逻辑两部分，看来也许是一个难题。

实际上，一旦在你的产品里有一台微型计算机，假定能使这“黑盒子”担任尽量多的工作，那末在经济上是极为有利的；这样你就必须检查是否还需要每个外部逻辑门。

我们知道，存储器是由一定数目的单元组成的。为了在微型计算机系统内扩大所实现的逻辑功能，你可以简单地编写一些附加的指令序列，把这些指令序列驻留在否则就要被浪费的存储器中。或者，需要增加程序存储器，但增加的费用很小。

而且，与数字逻辑研制的费用相比，微型计算机逻辑的研制时间短而花费小。一种典型的微型计算机系统的研制周期可以用第7页的图说明。

在上图所示的微型计算机研制周期中仍然有重复循环，但同研制数字逻辑相比，微型计算机研制周期中的重复循环所需要的时间较短，费用也较少。

每种微型计算机都是用一种开发系统研制出来的。这些开发系统的特性和操作，因不同的生产厂家而有所不同；然而，它们都具有以下功能：

1) 你可以模拟微型计算机系统，并使你确信没有必要制做一块试验板。