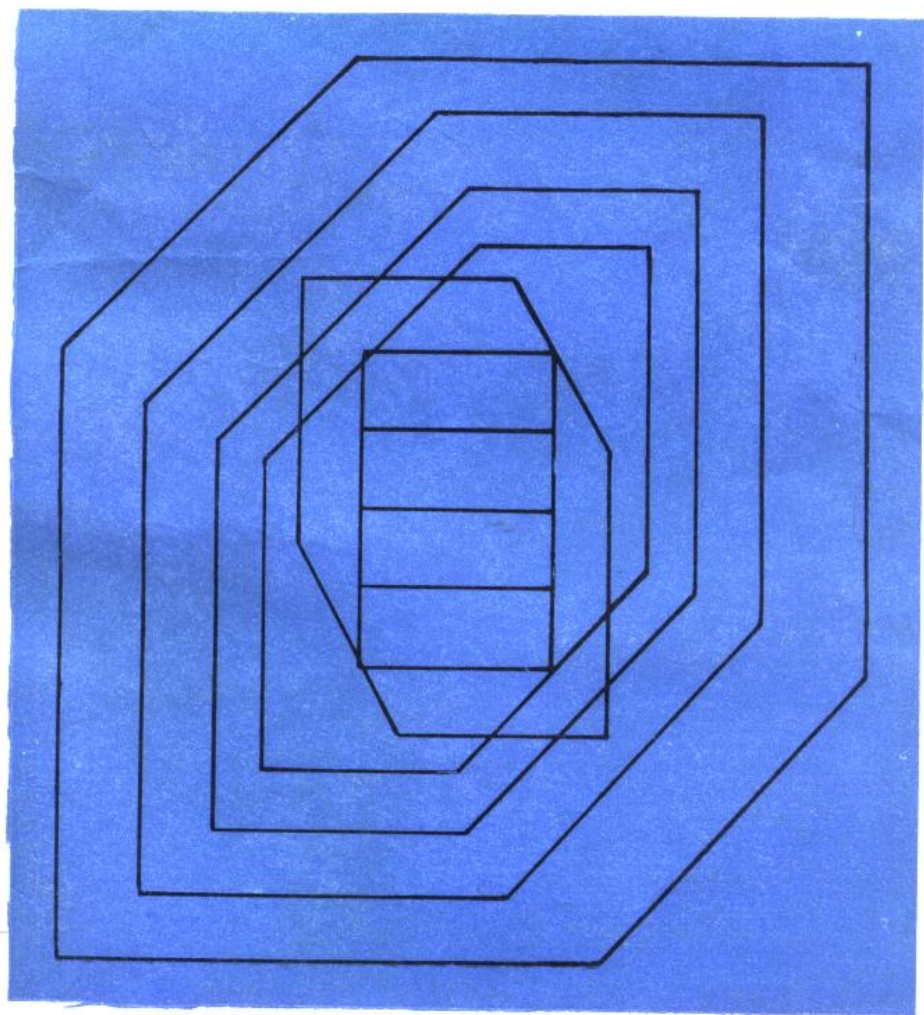


北京市高等教育自学考试用书

微机实验

夏业敏 编



北京市高等教育自学考试用书

微机实验

夏业敏 编

北京大学出版社

内 容 提 要

本书是为计算机软件专业成人自学考试编写的微机实验教材。全书共分九章。前六章介绍 Z80 的几个常用芯片, Z80 指令系统和数/模、模/数转换。后三章实验部分是: 基本程序设计实验、接口实验和选做实验。

本教材选用 EXERCISER-II 型单板机, 也可以选用 TP-801 型单板机。

本书是计算机软件专业高等教育成人自学考试的教材。也可供一般高等院校微机实验课参考。

北京市高等教育自学考试用书

微 机 实 验

夏业敏 编

责任编辑: 李采华

*

北京大学出版社出版

(北京大学校内)

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

*

850 × 1168 毫米 32 开本 10 印张 250 千字

1988 年 4 月第一版 1988 年 4 月第一次印刷

印数: 00001—7,000 册

ISBN7-301-00033-2/O-008

定价: 2.65 元

前 言

近年来以微型计算机为中心的一系列新技术得到了飞速发展和应用。微机由于具有经济效益高、实用性强、节能好、体积小、价格便宜、对环境要求较低而且易于普及等优点,因此在近十年内发展得很快,已逐渐取代小型机,并扩展到中型机领域。目前微型机已被用在数据处理、机器人、过程控制、计算机辅助设计等各个方面。

“微机原理”是一门实践性很强的课程,只有多实践,多做一些实验,才能真正掌握一些微机的原理,学会编程技巧并初步具有应用微机的能力。“微机实验”课,正是为此目的而设置的。本书针对计算机软件专业的特点,选编了一些程序量较大的实验,同时配有必要的接口实验。为了便于自学,每个实验的原理部分均作了较为详细的说明,并选用了 CPU 为 Z80 的单板机,这是国内广泛使用的一个机种,如 EXERCISER II, TP-801 等。

为了配合实验,原理部分介绍了 Z80 指令系统以及 Z80 的常用接口芯片等。

本书经北京大学计算机系张海藩和无线电系谢柏青二位副教授审阅,特别是张海藩同志反复、仔细地审阅了两遍,提出了许多宝贵的意见和建议。在此谨向他们表示衷心的感谢。

由于编者水平有限,缺点和错误在所难免,欢迎读者批评指正。

编 者

1986 年 2 月

目 录

第一章 Z80 CPU 的结构和时序	1
1.1 概述	1
1.2 Z80 CPU 的内部结构	3
1.3 标志寄存器和标志位	7
1.4 Z80 CPU 引脚功能	11
1.5 时序	13
习题	19
第二章 Z80 CPU 的指令系统	21
2.1 概述	21
2.2 数据的传送与交换指令	22
2.3 数据块的转移与搜索指令	29
2.4 算术和逻辑运算指令	31
2.5 循环和移位指令	35
2.6 位操作指令	39
2.7 转移、调用和返回指令	41
2.8 CPU 控制指令	47
第三章 输入、输出和中断	48
3.1 输入和输出	48
3.2 中断	54
习题	68
第四章 并行输入/输出接口芯片 Z80 PIO	69
4.1 概述	69
4.2 PIO 方框图和端口结构	71
4.3 PIO 引脚说明	73

4.4	初始化 PIO.....	77
4.5	应用举例	85
	习题	88
第五章	计数器-定时器芯片 Z80 CTC	89
5.1	概述	89
5.2	CTC 的内部结构	89
5.3	引脚说明	92
5.4	CTC 的工作方式	94
5.5	初始化 CTC	97
5.6	CTC 的中断	100
5.7	应用举例	102
	习题	108
第六章	数/模和模/数转换.....	109
6.1	概述	109
6.2	D/A 转换器	109
6.3	A/D 转换器	117
	习题	128
第七章	基本程序设计实验.....	129
7.1	Z80 E-II 型单板机简介	129
7.2	Z80 E-II 型单板机监控程序简介	132
7.3	键盘操作说明	138
7.4	汇编语言的源程序格式	146
	实验一 单板机的键盘操作	149
	实验二 不同进位制数之间的相互转换	158
	实验三 多精度十进制数的加减法	166
	实验四 乘法运算	173
	实验五 16 位二进制数的除法和开方运算	182
	实验六 排序	187
第八章	接口实验.....	194
8.1	端口地址	194
8.2	Z80 CTC 的通道安排	194

8.3	PIO, CTC 编程提要	197
8.4	逻辑实验箱和逻辑笔简介	198
	实验七 Z80 PIO 接口实验(一)	201
	实验八 中断	206
	实验九 Z80 CTC 的定时和计数	214
	实验十 Z80 PIO 接口实验(二)	216
	实验十一 交通信号灯自动控制	221
8.5	LED 七段显示器原理	226
	实验十二 LED 七段显示器	230
8.6	ADA-168 数模板	232
	实验十三 数/模(D/A)转换实验	234
	实验十四 模/数(A/D)转换实验	237
第九章 选做实验		244
9.1	EPROM 编程器	244
	实验十五 EPROM 编程	248
9.2	录音机的接口电路	250
	实验十六 录音机的转储与输入	253
9.3	DP 822打印机简介	254
	实验十七 打印机的使用	259
9.4	键盘输入译码程序	262
	实验十八 乐曲演奏程序	270
	实验十九 电子琴程序	273
	实验二十 电子时钟	276
	实验二十一 直流电机的调速	279
附录 1 Z80 指令的机器码表		283
附录 2 Z80 指令功能表		294
附录 3 标志位操作摘要		305
附录 4 集成电路引脚图		308

第一章 Z80 CPU 的结构和时序

1.1 概 述

CPU 是中央处理单元 (Central Processing Unit) 的简称, 由大规模集成电路 (LSI) 制成。常见的字长为八位的 CPU 有 Intel 公司的 8080A 和 8085A, Motorola 公司的 6800 等。字长十六位的 CPU 有 8086, Z8000 和 68000 等。此外 IBM PC 中选用微处理器 8088, 它的软件与十六位的 CPU 8086 兼容, 可支持十六位的乘法操作, 使系统具有较强的运算功能, 同时 8088 又具有八位数据总线, 可与通常为八位的输入/输出外设直接相接。

CPU 的主要功能是对数据进行运算、判断和传送, 以及对存储器、端口 I/O 和 CPU 本身进行控制。运算和控制的功能决定了它本身的地位, 它是微型机的中枢。

Z80 CPU 是 Zilog 公司 1976 年研制出来的。其特点如下:

1. 接单一的正 5 V 电源, 而在它以前的 Intel 8080A 要用三种电源。

2. 有 158 条指令, 其中包括 8080A 的 78 条指令, 软件和 8080A 兼容。

3. 指令功能较强, 能对字、字节、位和字符串进行操作; 能对数据块进行检索和传送。在寻址方式中设有相对寻址和变址寻址, 这些都使得它具有很强的数据处理能力 (具体内容将在下一章讲述)。

4. 在 CPU 内部设有可交换的通用寄存器组、标志寄存器、I 和 R 寄存器等。

5. 在响应中断的方式上也有一些特色, 并设有不可屏蔽中断。

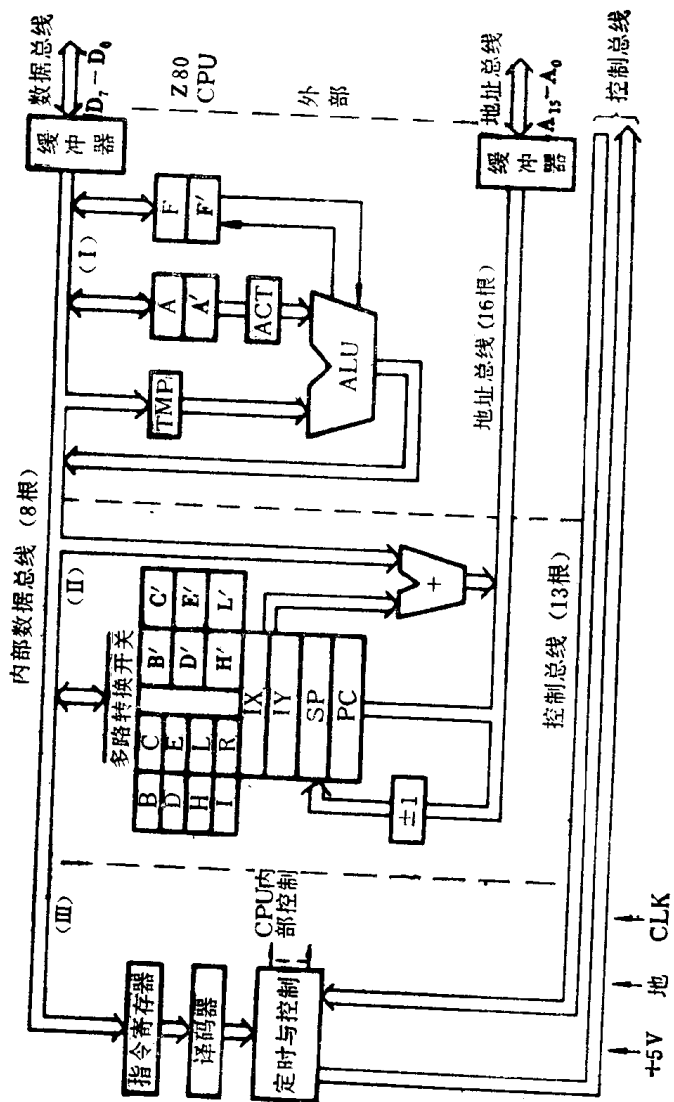


图 1-1 Z80 CPU 的结构框图

这些将在第六章详细介绍。

Z80 CPU 按时钟频率可分为四档: Z80H, 8MHz; Z80B, 6MHz; Z80A, 4MHz; Z80, 2.5MHz。

1.2 Z80 CPU 的内部结构

一、Z80 CPU 的内部结构及框图

Z80 CPU 的内部结构虽然较为复杂,但从其结构框图 1-1 来看,主要分为三部分: 算术逻辑单元 ALU (虚框 I, 包括累加器 A 和标志寄存器 F); CPU 定时与控制 (虚框 III, 包括指令寄存器和译码器); 寄存器阵列 (虚框 II)。这三部分通过内部总线连接起来。CPU 芯片对外的连接,也分为三部分: 经由数据总线缓冲器输入输出的数据总线 (D_0-D_7), 经由地址总线缓冲器输出的地址总线 (A_0-A_{15}) 以及与 CPU 定时和控制部件相连的控制总线 (13 根)。此外还有电源、地线和时钟线的连接。

从编程角度出发,只要对 CPU 的内部结构有一般性的了解,其中的寄存器阵列,由于在编程过程中总要用到,应熟悉其功能与特点。下面的介绍也将侧重在寄存器阵列部分,并在后一节里专门介绍标志寄存器。

二、Z80 CPU 内部寄存器

Z80 CPU 中共有 208 位静态 RAM 读/写存储器单元 (寄存器) 提供程序员使用。这些寄存器包括两组六个通用寄存器,它们可以单独作为八位的寄存器用或成对地作为十六位的寄存器用。另外还有两组累加器和标志寄存器。

Z80 CPU 中还包括四个十六位的专用寄存器,即堆栈指示器、程序计数器和两个变址寄存器。此外还有一个刷新计数器和一个中断寄存器。

Z80 CPU 的寄存器阵列图如图 1-2 所示。除了上边介绍的通用和专用寄存器组之外,右下一组为两个中断状态触发器和一

对中断方式触发器,它们的功能将在第三章中介绍。

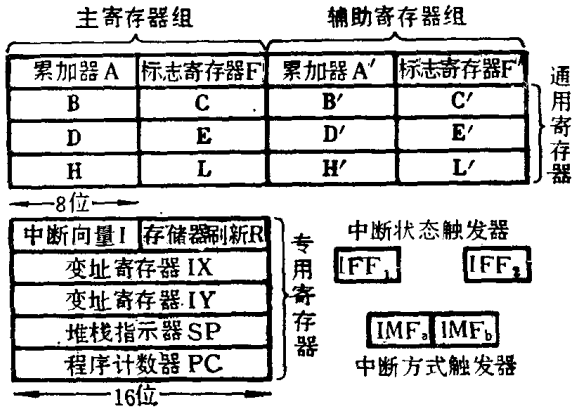


图 1-2 Z80 CPU 的寄存器阵列图

1. 通用寄存器

有两组相对应的通用寄存器,每组包括六个八位的寄存器。一组叫主寄存器 (Main Register Set), 其中有 B, C, D, E, H, L 六个寄存器。另一组叫辅助寄存器 (Alternate Register Set), 其中有 B', C', D', E', H', L' 六个寄存器。

通用寄存器的主要功能是暂存参加运算的操作数及其中间结果。当存储十六位地址或立即数时,可按 BC, DE, HL 组合起来使用,习惯上常把 HL 寄存器对当作地址指针,把 BC 寄存器对当作计数器用。

Z80 CPU 的指令系统中,指令的操作均在主寄存器之间或主寄存器与存储器之间进行,辅助寄存器中的内容,要先用“交换指令”(见下一章)调入主寄存器后,方能进行传送、运算等操作。因此可将主寄存器中的内容暂存到辅助寄存器,灵活地使用寄存器,不必访问存储器,以节省时间。在子程序和中断服务程序中,也常用交换指令把主寄存器中的内容暂存到辅助寄存器中保存起来,这样使用十分方便。

2. 累加器 (Accumulator) A

累加器是一个八位的寄存器, 它要存每次运算(算术或逻辑)的操作数或中间结果, 并通过它把运算的结果或数据经 I/O 端口芯片送至外设。累加器通过 ALU 可以实现移位、循环等操作。Z80 CPU 指令系统中很多指令均和它有关。

3. 专用寄存器

(1) 程序计数器 (Program Counter) PC

程序计数器是十六位的计数器, PC 中的内容就是即将要执行的下一条指令的地址。CPU 将 PC 的内容送到地址总线之后, PC 就自动加 1。

正是 PC 计数器的这一特性, 使得取出存储器中的某一个地址单元中的指令之后, 能够顺序地取出下一个单元中的内容, 从而顺序地执行完程序。如果在执行程序中, 需要转移, 则应将转移地址送入 PC。

(2) 堆栈指示器 (Stack Pointer) SP

所谓堆栈就是在存储器 RAM 中开辟一个区域, 存放暂时不用的数据或地址, 最先存入的数据或地址构成了“栈底”, 最后存入的数据或地址构成了“栈顶”。存入数据可通过压入 (PUSH) 指令, 取出数据则可用弹出 (POP) 指令。数据或地址从特定的 CPU 寄存器压入堆栈, 或从堆栈弹出到 CPU 寄存器, 是按照“后进先出”的方式, 即最先弹出的数据总是最后一个压入的数据。进栈或出栈均需通过栈顶进行。每一次压入或弹出操作, 栈顶地址均随之改变。在 Z80 CPU 中设置一个十六位寄存器 SP, 就是用来指示上述随时变化的栈顶地址。

(3) 变址寄存器 (Index Register) IX 和 IY

IX 和 IY 都是十六位的寄存器, 这两个寄存器的作用完全相同。它们为变址寻址方式提供所需要的十六位基地址。

(4) 中断向量寄存器 (Interrupt Vector Register) I

I 是一个八位寄存器，用来存放中断向量的高八位地址。将在第三章的中断一节详细介绍。

(5) 存储器刷新计数器 (Memory Refresh Counter) R

R 是一个七位计数器，每取完一次指令操作码后自动加 1。R 计数器为动态存储器提供刷新地址。

三、其它部分

1. 算术逻辑单元 (Arithmetic and Logic Unit) ALU

CPU 的八位算术逻辑指令是在 ALU 中执行的。ALU 还控制内部寄存器和外部数据总线之间的数据交换。ALU 的主要功能如下：

- (1) 算术运算：加、减、加 1、减 1。
- (2) 逻辑运算：与、或、异或。
- (3) 其它操作：移位、循环。位置位、位复位、位测试。

算术逻辑单元 ALU 部分的结构框图如图 1-3 所示。若有两个数进行加法运算，则其中一个数经累加器 A 送至 ALU，而另一个数则由内部总线经暂寄存器送至 ALU。运算结果由 ALU 再经内部数据总线回送到累加器 A。同时根据运算结果置标志寄存器的相应标志位。

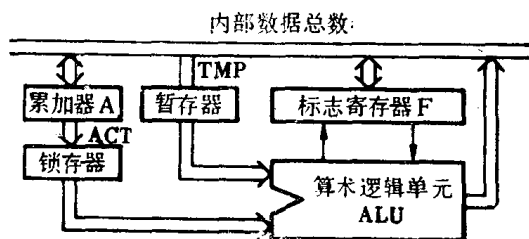


图 1-3 ALU 部分的结构框图

2. 指令寄存器 IR 和 CPU 控制

ALU 是运算器，这部分则相当于控制器。其工作过程如下：

取出指令(从存储器 RAM 中)→指令寄存器 IR→指令译码→

发出全部控制信号。(控制外部芯片或 CPU 本身。)

1.3 标志寄存器和标志位

标志寄存器是与算术逻辑单元 ALU 直接相连的一个八位寄存器,它包括四个可测试的标志位和两个不可测试的标志位,是我们窥测 ALU 操作结果的某种状态的唯一窗口。因此了解这些标志位将有助于今后的程序调试工作。

Z80 CPU 标志寄存器中的六个标志位的排列如下:

D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
S	Z	X	H	X	P/V	N	Cy

一、可测试标志位

所谓可测试标志位,就是可以通过程序来检查这几个标志位,从而作为转移、调用或返回等指令的条件。四个可测试标志位是:

1. 符号标志位 (Sign) S

符号标志位用于带符号数的运算。如果操作结果为负则置 S 位为 1; 反之,则置 S 位为 0。在八位微型机中,用最高有效位 D₇ 代表数的符号,负数的 D₇ 位为 1,所以这个标志位存放了累加器中 D₇ 位的状态。

2. 零标志位 (Zero) Z

若运算结果为零,则将 Z 标志位置 1; 否则将此标志位置 0。这对于算术运算和逻辑运算都适用。

3. 奇偶/溢出标志位 (Parity/Overflow) P/V

用于逻辑运算(与、或、异或)结果的奇偶性。如果运算结果中有偶数个 1,则置 P/V 标志位为 1; 反之,置 P/V 位为 0。在做算术运算时,如果发生溢出,则此标志位置 1; 否则置 0。

4. 进位标志位 (Carry) Cy

(1) 两数相加后在最高位产生进位,则置进位位 Cy 为 1; 反

之，则置此位为 0。

(2) 两数相减时，最高有效位需要借位，则置进位位 Cy 为 1。

(3) 执行循环或移位指令时，也可能改变进位位。

(4) 根据程序需要，可以直接用程序使进位位置 1 或置 0。

二、不可测试标志位

1. 半进位标志位 (Half Carry) H

在二-十进制 (BCD 码) 运算中，半进位标志位 H 是低四位向高四位产生的进位或借位。当执行十进制调整指令 (DAA) 时，用这个标志位来修正 BCD 码加法或减法的结果。

2. 加减标志位 (Add/Subtract) N

如果前一运算是减，则 $N = 1$ ；否则 $N = 0$ 。H 和 N 标志位和十进制调整指令 DAA 联用，以便在 BCD 加或减之后修改 BCD 结果。

三、置标志位举例

1. 算术运算或逻辑运算的结果置标志位

$$(1) (+15) + (+48) = +63 \quad (2) (+15) \vee (+48) = +63$$

$$\begin{array}{r} 00001111 \\ Cy 00110000 \\ \hline \boxed{0} 00111111 \end{array}$$

$$\begin{array}{r} 00001111 \\ Cy 00110000 \\ \hline \boxed{0} 00111111 \end{array}$$

运算结果置各标志位：

$$\left. \begin{array}{ll} S = 0 & D_7 \text{ 位为 } 0, \text{ 结果为正} \\ Z = 0 & \text{结果不为 } 0 \\ Cy = 0 & \text{运算结果, 最高有效位无进位} \end{array} \right\} \text{解释同左}$$

$$P/V = 0 \quad \text{结果没有产生溢出} \quad P/V = 1 \quad \text{运算结果有偶数个 } 1。$$

从上例看出：

① P/V 标志位在算术运算和逻辑运算中意义不同。

② Cy 标志位在逻辑运算后总是为 0，故程序中常通过执行逻辑指令把进位位 Cy 清 0。

2. 进位标志位 Cy 和溢出标志位 P/V

(1) $(+120) + (+105) = +225$

$$\begin{array}{r}
 01111000 \\
 \text{Cy} \quad +) 01101001 \\
 \hline
 \boxed{0} \quad 11100001
 \end{array}$$

和数的补码为 11100001

和数为 $10011111 = -31$ (结果是错的)

两个正数相加，结果为负，显然是错的，原因是运算结果超出了八位带符号数的补码范围 $(+127 \sim -128)$ ，产生了溢出。

运算结果置标志位：Cy = 0, P/V = 1。

(2) $(-15) + (-16) = -31$

$$\begin{array}{r}
 11110001 \quad (10001111 \text{ 的补码}) \\
 \text{Cy} \quad +) 11110000 \quad (10010000 \text{ 的补码}) \\
 \hline
 \boxed{1} \quad 11100001
 \end{array}$$

和数的补码为 11100001

和数为 $10011111 = -31$

运算结果置标志位：Cy = 1, P/V = 0。

(3) $(-120) + (-105) = -225$

$$\begin{array}{r}
 10001000 \quad (11111000 \text{ 的补码}) \\
 \text{Cy} \quad +) 10010111 \quad (11101001 \text{ 的补码}) \\
 \hline
 \boxed{1} \quad 00011111 \quad (+31)
 \end{array}$$

两个负数相加，结果为正数，显然是错误的，因为运算结果 < -128 ，超出了八位带符号数的补码范围，运算结果置溢出标志位 P/V = 1。现对 Cy 和 P/V 标志位小结如下：

① 只有当两个同号数相加，或是两个异号数相减时，才可能产生溢出。

② 溢出的原因是运算结果超出了八位带符号数的补码范围。

③ 溢出时, 溢出标志位 $P/V = 1$ 。

④ $C_y = 1$ 或 $C_y = 0$ 均可能产生溢出, C_y 和 P/V 是两个性质完全不同的标志位。

3. 两个同符号整数相减, 若为:

(1) 小数减大数, 则相减结果为负数 ($S = 1$), 且应该有借位 ($C_y = 1$)。

(2) 大数减小数或两数相等, 则相减结果为正数 ($S = 0$), 且无借位 ($C_y = 0$)。

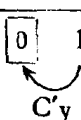
Z80 CPU 做此减法的步骤是:

① 求出减数的补码, 和被减数做加法。

② 取进位的反码送进位标志位 C_y 。

例 1 $3 - 6 = -3$

$$\begin{array}{r}
 00000011 \\
 + 11111010 \quad \leftarrow -6 \text{ 的补码} \\
 \hline
 \boxed{0}11111101
 \end{array}$$

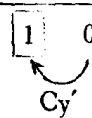


∴ 结果: 对 11111101 求补为: $10000011 (-3)$

$$C_y = \overline{C'_y} = 1 \text{ (小数减大数有借位)}$$

例 2 $6 - 3 = 3$

$$\begin{array}{r}
 00000110 \\
 + 11111101 \quad \leftarrow -3 \text{ 的补码} \\
 \hline
 \boxed{1}00000011
 \end{array}$$



∴ 结果: $00000011 (+3)$

$$C_y = \overline{C'_y} = 0$$