

单片机

开发手册



华龄出版社

单片机开发手册

谭运光 陈安源
熊正明 华 健 金 涛 编著

华 龄 出 版 社

(京)新登字068号

责任编辑 华 健

封面设计 金 涛

图书在版编目(CIP)数据

单片机开发手册. —北京: 华龄出版社, 1994. 12

ISBN 7-80082-380-6

I. 单… I. Ⅱ. 单片微型计算机—开发—手册 IV. TP3 68.1-62
中国版本图书馆 CIP 数据核字 (94) 第 14567 号

单片机开发手册

谭运光 陈安源 熊正明 华 健 金 涛 编著

出版发行 华龄出版社

(北京市东城区小乘巷 21 号 邮编 100035)

经 销 新华书店

北京印刷学院实习厂

787×1092 毫米 1/32 23.5 印张 601 千字

1994 年 12 月第一版 1994 年 12 月第一次印刷

印数 1-3000 册

ISBN 7-80082-380-6/TP·4

定价: 58.00 元

前 言

单片机于 1976 年问世之后,得到广泛的应用,尤其在工业控制、自动化仪器、仪表、计算机接口、智能化设备等应用领域独领风骚。正是因为其具有价廉物美、功能强、体积小、使用灵活方便等特点,故对于目前我国国情特别行之有效。

目前,国内从事单片机应用开发人员越来越多,而且大多成绩斐然,给社会带来巨大的经济效益,同时大家又都将自己的实践经验经过系统的整理编成书籍,为单片机的应用在中国普及与发展添了砖加了瓦,形成一种单片机应用进入良形循环的大好景象。然而,美中不足的是,在书店买单片机方面书籍的时候,不难发现几乎所有单片机的书籍都是介绍 MCS-51 系列单片机,这样对于初学者与设计者都极为不利,因为初学者应该知道更多更新单片机芯片的特性;设计者应该在系统设计过程中挑选更合适的单片机芯片。所以本书第一篇主要介绍各种单片机芯片,以供大家参考

第二篇主要介绍设计过程中的分析工作,包括总体设计、硬件设计和软件设计分析。第三篇介绍的是大家普遍采用的 SICE 通用单片机开发系统。

本书的编写过程中,得到华龄出版社的孙涛、中科院计算所董立忠以及其他有关专家的大力支持,在此表示由衷的感谢!

由于编写时间紧迫,加之编者水平有限,本书不足之处在所难免,恳请大家多提宝贵意见。

编著者

1994 年 7 月于北京

目 录

第一篇 单片机芯片介绍

第一章	KS56C821 单片机	(1)
第二章	SMC6200 系列单片机	(37)
第三章	PIC16C5X 系列单片机	(85)
第四章	MCS-51 系列单片机	(118)
第五章	MC68HC11 系列单片机	(158)

第二篇 设计方法分析

第一章	设计准备工作	(201)
第二章	硬件设计	(202)
第三章	软件设计	(244)
第四章	设计实用技术及实例	(264)

第三篇 SICE 通用单片机开发系统

第一章	开发系统概述	(331)
第二章	SICE 通用单片机开发系统概述	(333)
第三章	SICE 仿真特性	(337)
第四章	EPROM 固化器及输入、输出设备	(338)
第五章	SICE 系统与系统主机的连接	(344)
第六章	编辑方法	(345)
第七章	汇编与反汇编	(347)
第八章	交叉汇编及传送	(350)
第九章	SICE-51 系统操作命令	(351)

附录一	本书所用集成电路芯片引脚图	(364)
-----	---------------------	-------

附录二	设计任务书书写格式	(371)
-----	-----------------	-------

附录三	参考书籍	(372)
-----	------------	-------

第一篇 单片机芯片介绍

对于计算机开发人员来说,首先应该熟练操作PC机,也就是说大家都应能轻而易举地列出计算机的主要构成部分:微处理器、存储器、并行、串行I/O端口、A/D、D/A转换器等。当将各构成部分全部放在一块电路板上时称为单板机,如Z80等,而随着现代科学技术的高速发展,人们不再满足单板机,而是利用超大规模集成技术(VLSI)将各构成部分全部集成到一小块芯片上,这就是人们常说的单片机。

由于单片机具有体积小、功能齐全、存储容量大、低功耗等特点,在各个应用领域中都被广泛采用,特别是成本低,因而在中国具有广泛的市场,如许多电脑控制的洗衣机都是使用单片机代替原有的模拟和逻辑控制电路;还有不少空调生产厂家也正在开发电脑控制器代替原有的线路,控制器当然也是采用单片机。

正是因为单片机特别适合中国国情,所以目前各领域的设计师早已将单片机应用到自己熟悉的领域,并且成果非凡。然而,当设计师开始从事项目设计的时候,不管怎样选用单片机芯片,从来就是毫不犹豫,肯定是MCS-51系列单片机,因为他们对51系列单片机太熟悉了,其实从另外一个角度来看,就是对其他种类单片机芯片了解太少。假如了解了其他种类单片机,综合考虑各方面因素后,那么就不一定采用MCS-51系列单片机。本书将介绍国内市场出现的各种单片机芯片。

虽然单片机经历了4位机、8位机、16位机的发展,目前还出现了32位单片机,但是由于他们具有各自的独特功能及特点,因而同样都被广泛采用。4位单片机本书介绍EPSON公司生产的SMC6200系列单片机,三星公司生产的KS56C821单片机。8位单片机主要介绍MCS-51系列、PIC16C5X系列、MC68HC11系列单片机。这里需要说明的是,本书不能不介绍MCS-51系列单片机,这是因为:第一,与其他种类单片机放在一起介绍便于比较;第二,设计和开发系统的介绍主要以51系列单片机为依据,便于查找。

在项目设计的开端,最主要的工作是选用单片机芯片,当然最为关心的是芯片的CPU特点,ROM容量和RAM容量的大小,I/O端口的数目和特点等。因此,为了方便单片机芯片各方面的比较,本书主要从以上几个部分介绍各种单片机芯片。

指令系统是表征一种单片机性能的重要标志。因为指令的利用效率决定了单片机系统的诸多指标,假如一个简单的功能需要很长的程序指令,执行它时肯定长期占有CPU时间,严重影响系统执行速度,这种单片机可能成本低但绝不能用于高速系统。所以,设计之前了解各单片机的指令系统也是不可忽略的因素。

第一章 KS56C821 单片机

第一节 产品概述。

KS56C821 是用 CMOS 工艺制作的 4 位单片机, 它使用 SMCS-56 为中心 CPU, 包含 RAM、ROM, 36 条 I/O 线, 可编程定时器/计数器, 蜂鸣器输出, 充足的 LCD 点阵, 和段驱动脚。KS56C821 可在不同应用领域中用于控制功能, 并且特别适合于设计多用数据和 LCD 游戏。该器件的基本结构见图 1.1 (a), 图 1.1 (b) 是其管脚图。

特点:

- RAM: 数据: 5120×4 位
显示: 108×5 位
- ROM: 8192×8 位
- 存储映象 I/O 结构
- 一个 8-比特基本定时器
- 一个 8-比特定时器/计数器
- 能够选择 3 种内部时钟之一, 或子系统时钟
- 秒表定时器
- 多种中断源
 - 三种外部向量中断: INT0, INT1, INTP0
 - 两种内部向量中断: INTB, INTP
 - 两种准中断: INTW, INT2
- 36 条 I/O 线
 - 输入: 4 个脚
 - I/O: 16 脚; 内部 LED 驱动
 - 输出: 最多 16 个 1-比特电平输出脚 (共享段驱动输出)
 - N 沟道源极开路型 I/O; 8 个脚; 能够处理达 9 种电压, 单独上拉电阻可屏蔽选择
- LCD 显示
 - 12 种特性点阵显示 (5×7)
 - 12 位显示 (8 段)
 - 60 段和 9 公共脚
- 2kHz 或 4kHz 可选蜂鸣输出
- 振荡电路
 - 主系统时钟使用软件可选内部 RC 振荡电路
 - 子系统时钟使用 32.768kHz 晶体振荡电路
- 两类降低功耗方式
 - IDLE: 仅 CPU 时钟停止

- STOP: 主系统时钟停止
- 为保持功耗, 不同的指令周期。
 - 2.8, 5.5, 44.1 μ S 在 1.45MHz 时, 主系统时钟
 - 122, 244, 1952 μ S 在 32.768kHz 时, 子系统时钟
- 工作电压范围: 2.2V~6.0V
- 100 QFP-N 封装

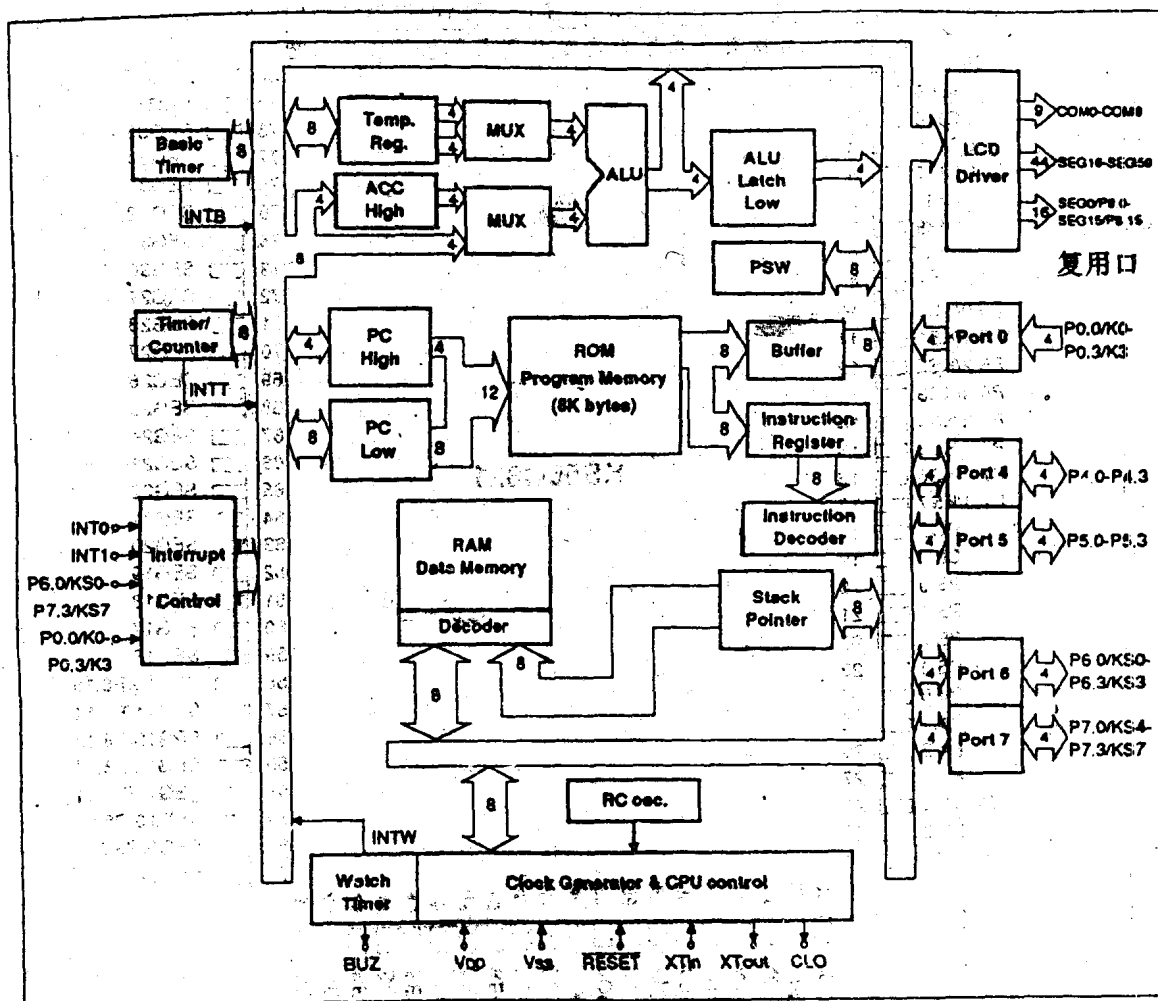


图 1.1 (a) 结构框图

第二节 CPU 功能

一、程序计数器 (PC)

程序计数器是一个二进制计数器,它用来保存下一条被执行指令的程序地址。PC 包含 13 位,因此 KS56C821 存储器范围是 0000H~1FFFH。

当产生 $\overline{\text{RESET}}$ 信号时,程序存储器地址的高 5 位 PC_{12-8} 的内容变成 0000H,并且另外 8 位 PC_{7-0} 的内容设为 0001H。这时 PC 的内容由 $\overline{\text{RESET}}$ 向量地址的设定值初始化。

当中断发生时,PC 重新复位且 PC 的内容由相关的中断向量地址设定(程序存储器空间: 0002H~000BH)。

通常,PC 由被取指令字节的数字递增,当执行调用或跳转指令时,有效地址的内容被装入 PC。不管怎样,当执行 REF 指令时,PC 不递增。执行完 REF 指令之后,程序从 REF 指令的下一个地址开始。

二、程序存储器 (ROM)

程序存储器是一个能够保存程序或查表数据的存储器。KS56C821 的程序存储器的地址范围有 8k 字节(地址: 0000H~1FFFH)。程序存储器分布图见图 1.2,查表数据能被查表指令(LDC)处理。

000H	向量地址区
000BH	
000CH	通用 ROM 区
001FH	
0020H	REF 指令参照区
007FH	
0080H	通用 ROM 区
1FFFH	

图 1.2 程序存储器分布图

程序存储器分成以下三个部分：

①地址：0000H~000BH

这是RESET的向量表区域或中断向量地址。这个区域存放着中断服务程序的入口地址，也是用来设置对应中断的初始化条件的 EMB 和 ERB 的值。当一个中断向量地址区域不是用作中断时，相应区域可用作通用的 ROM。图 1.3 显示相应中断的向量地址。图 1.4 显示一个向量地址的内容。

	7	6	5	4	3	2	1	0	
000H	复位向量地址								
0002H	基本定时向量地址								
0004H	INT0 向量地址								
0006H	INT0 向量地址								
0008H	INTP0 向量地址								
000AH	定时/计数向量地址								

图 1.3 RESET和中断的向量地址

EMB	ERB	0	PC12	PC11	PC10	PC9	PC8
PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0

图 1.4 一个向量地址的内容

②地址：0020H—007FH

这是 REF 指令参照区，这个区域不能用作通用 ROM 区域。

REF 指令允许 1—字节表示 2—字节、3—字节指令，或两个 1—字节指令，从而减少程序步的数字。

③地址：000CH—001FH 和 0080H—1FFFFH

这是通用的程序存储器区域。

三、数据存储器 (RAM)

数据存储器是一个暂时保存处理中数据的存储器，它包含 5120×4 —比特数据 RAM 和 108×5 —比特显示 RAM (地址: 100H—1BBH)。数据存储器分布图见图 1.5。不管怎样，它能由 1, 4 和 8 比特寻址指令处理。对于 8—比特存储器寻址，在操作数中必须指示数据存储器的偶数地址。显示数据存储器区仅能够被 8—比特单元寻址处理。

数据存储器可分为 3 个库 (BANK0, BANK1 和 BANK15)，并且任一库可被选为工作数据存储器，方法是使用存储器选择指令 (SMBn) 和存储器使能标记的状态 (EMB)。(指的是寻址模式)。数据存储器的复位值未定，建议复位时设定它的初始化值。

• BANK0 (地址: 000H—0FFH)

最低 32 个半字节 (地址: 002H—01FH) 可用作工作寄存器，另外 224 个半字节 (地址: 020H—0FFH) 能分成堆栈区和通用数据存储器，堆栈区用于补充子程序调用和返回，以及中断处理。

• BANK1 (地址: 100H—1FFH)

BANK1 有 20 页数据存储器，前 19 页用于通用数据存储器由 256×4 位构成，而第 20 页用于 LCD 显示数据存储器包含 108×5 位。KS56C821 专门使用一个 BANK1 页选择寄存器 (PASR) 用于选择这 20 页中的一页。

BANK1 页选择寄存器 (PASR)

PASR 是一个用于选择 BANK1 某页的 5—比特只写寄存器，并且映象于 RAM 地址 OF90H，它仅被 8 位 RAM 控制指令改写；而且 SMB3 位是 0。

PASR 在不变时尽可能长时间保持原值，而且复位值为 0。所以，当从别的库 (BANK0 或 BANK15) 回到 BANK1，而且 PASR 的内容改变时，则选择 BANK1 上次指定的页。

PASR 在中断服务程序中不必变化，因为当程序结束时它的值也不能被原始值覆盖掉。

(例) 当选择 BANK1 的 13H 页当作工作数据存储器

SMB15

LD EA, #13H

LD PASR, EA

SMB1

• BANK15 (地址: F80H—FFFH)

KS56C821 使用一个存储器映象 I/O 结构，将每个 RAM 地址对应的硬件分配到这个区域。

四、寄存器

1. 工作寄存器

工作寄存器可用于暂时保存用于计算的立即数或保存用于间接寻址方式的指针值，寄存器结构见图 1.6，而且它们映象于 RAM 地址 000H—01FH。

寄存器由 4 个库构成。任一库可被选为工作寄存器库。方法是使用寄存器库选择指令 (SRBn) 和寄存器库使能标记的状态 (ERB) (参考寻址模式)

8 个 4 位总体寄存器，Y, Z, W, X, N, L, E 和 A，不仅能用 4 位操作，而且能构成 8 位对寄存器 EA, HL, WX, YZ 和 WL。当寄存器用作 8 位对寄存器时，寄存器 A, L, X 和 L 变成低半字节。

4—比特 寄存器 $\times 8 \times 4$ 库

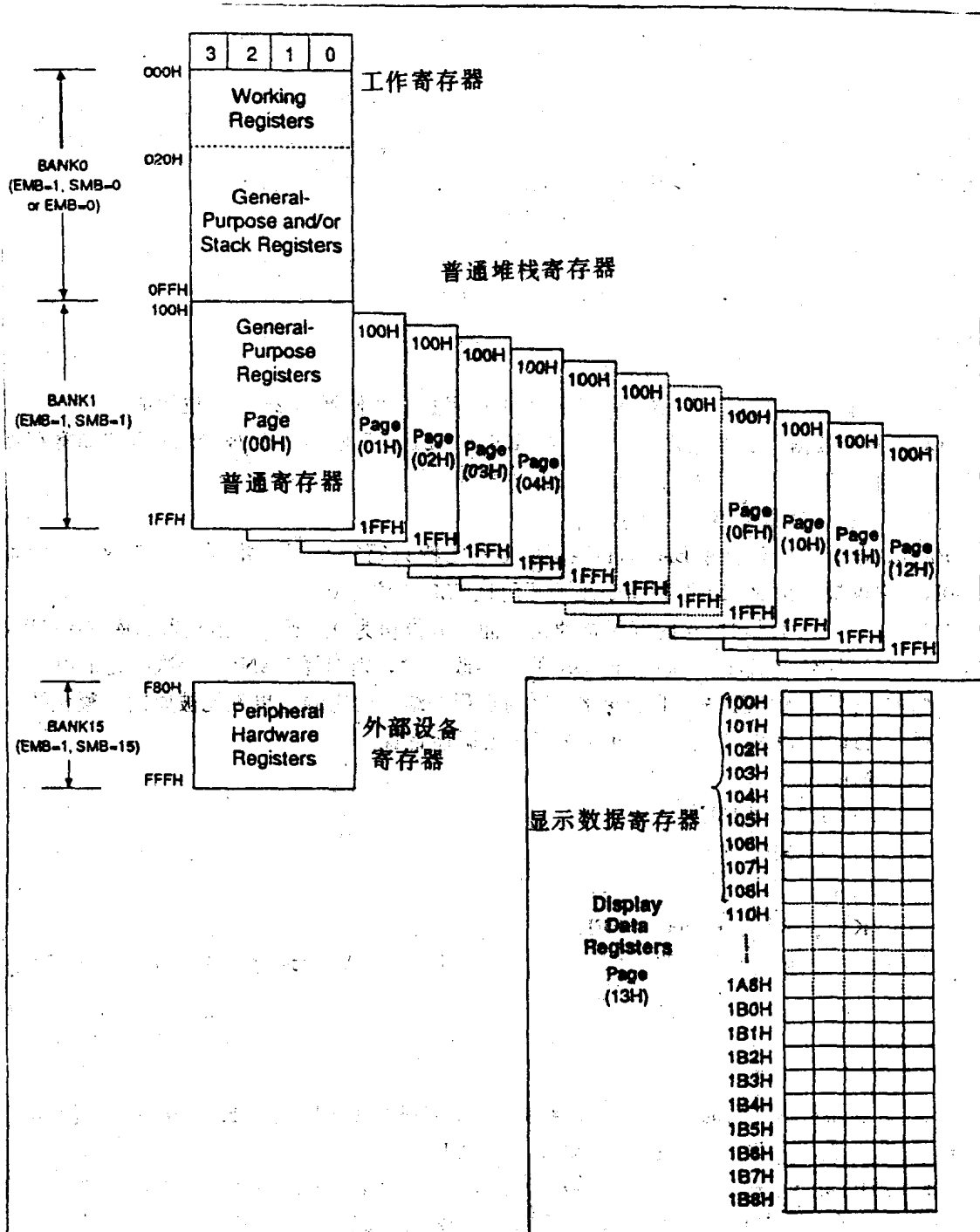


图 1.5 数据存储分布图

000H	A	寄存器 BANK0 (ERB=0, 或 ERB=1 和 SRB=0)
001H	E	
002H	L	
003H	H	
004H	X	
005H	W	
006H	Z	
007H	Y	
008H	BANK1	寄存器 BANK1 (ERB=1, SRB=1)
00FH	(同 BANK0)	
010H	BANK2	寄存器 BANK2 (ERB=1, SRB=2)
017H	(同 BANK1)	
018H	BANK3	寄存器 BANK3 (ERB=1, SRB=3)
01FH	(同 BANK0)	

图 1.6 寄存器分布图

8—比特 寄存器×4×4 库

当工作在间接寻址模式时，8 位对寄存器 WX, WL 和 HL 可用作一个指定存储地址和比字位置的数据指针。仅当 8 位对寄存器 HL 用于一个数据指针操作时，它可对 L 寄存器进行自动递增和递减。(LDI A, @HL; LDD A, @HL; XCHL A, @HL; XCHD A, @HL)

总体来说，BANK0 用于一个主程序，BANK1—BANK3 用于中断服务程序。所以，数据的混用可避免因寄存器引起的混乱。当超过 4 个中断时，在执行中断服务程序前使用同一个寄存器库将寄存器的内容通过 PUSH RR 指令存于堆栈区，即可避免寄存器数据丢失。并且后来可通过 POP RR 指令恢复寄存器数据。

2. 库选择寄存器 (SB)

库选择寄存器 (SB) 用于指定存储器库和寄存器库，而它包括寄存器库选择寄存器 (SRB) 和存储器库选择寄存器 (SMB)，SRB 和 SMB 的内容由 SRBn 和 SMBn 指令指定。当要保存和恢复用于中断和子程序调用的寄存器的内容时，使用 PUSH SB 和 POP SB 指令。

寄存器库选择寄存器 (SRB)

SRB 是一个指定一个寄存器库的 2—比特寄存器, 用 ERB 和 SRB_n (n=0, 1, 2, 3) 指令的状态可选择 4 个寄存器库中任意一个。

当不需要改变时, 其值可尽可能长时间保持, SRB 的复位值为 0 (参考寻址模式)

存储器库选择寄存器 (SMB)

SMB 是一个可选择三个数据存储器库其中之一的 4—比特寄存器, 它决定 12—比特数据存储器地址的高 4 位, 用于规定寻址范围。数据存储器的寻址范围由 EMB 和 SMB_n (n=0, 1, 15) 指令的状态指定。

当不需要改变时, 其值可尽可能长时间保持, SMB 的复位值为 0 (参考寻址模式)。

五、堆栈指针 (SP)

在执行一条调用、中断或压栈指令前, 堆栈区暂时保存当前程序 (PC、EMB、ERB、PSW) 的数据, SP 是一个指示保存当前数据的堆栈区的地址的 8—比特寄存器。不管 EMB 和 SMB 的值, 数据寄存器 BANK0 的地址 000H—0FFH 都可用作堆栈区。而且堆栈是一种 LIFO (后进先出) 类型。

SP 执行 PUSH 操作时递减, 而执行 POP 操作时递增, 堆栈指针总是表示存在堆栈顶端最近内容的顶端地址。

SP 映射到 RAM 地址 F80H—F81H, 可通过 8—比特 RAM 控制指令读写。不管怎样, SP 的位 0 必须设置为 0。

所以, SP 的复位值未定, 建议将 SP 的初始值设置为 0, 这样可以使用数据存储器 BANK0 的 RAM 地址 0FFH 当作堆栈区。

(例) SP 初始化为 00H

SMB 15 ; SMB←15 or BITR EMB

LD EA, #02 ; EA←02H

LD SP, EA ; F80H←A, F81H←E

在堆栈区, 调用一个子程序占用 4 个半字节, 而一个中断使用 6 个半字节。当子程序连续嵌套或不断使用中断时, 取出在执行子程序和中断中的上面描述的半字节数目计数, 用来按照子程序嵌套级的最大数目设定堆栈区, 而在这个范围内禁止使用数据。如果堆栈区和通用数据处理区共享, 保存的信息会丢失。

图 1.7 和图 1.8 展示数据如何存入堆栈和从堆栈中取出。

六、程序状态字 (PSW)

程序状态字 (PSW) 是一个表示程序执行状态的 8—比特寄存器。图 1.9 展示它的标记和比特分布。

PSW 映射于 RAM 地址 FB0H—FB1H, 而且不管 EMB 是何值在 1—比特直接寻址模式下都可执行 1—比特读/写。通过 1 或 4 比特读/写和 8 位读指令可改变 PSW。不管怎样, 在 RAM 地址 FB1H 中的 SC0, SC1 和 SC2 仅仅能够通过 8 位读指令处理。

当产生 RESET 信号时, EMB 和 ERB 被置为 RESET 向量地址中的值, 而进位标记的内容未知, IS0, IS1, SC0, SC1 和 SC2 复位值清零。

1. EMB (存储器库使能标记)

EMB 是一个 1—比特标记, 它通过改变数据存储器地址的高 4 位来指定 RAM 寻址范围, 而且映射于 RAM 比特地址 FB0H.1。

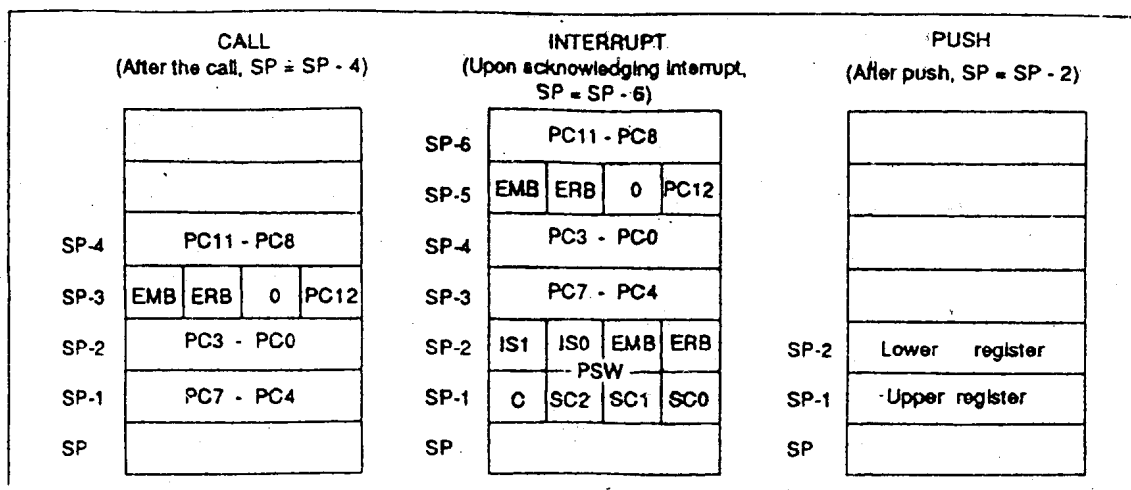


图 1.7 数据存入堆栈

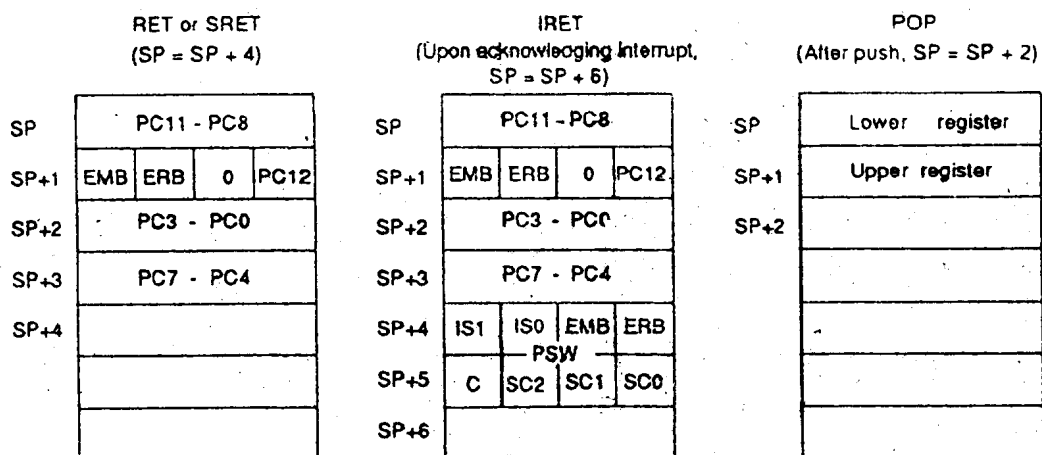


图 1.8 数据从堆栈取出

ERB	0	寄存器库禁止模式
	1	寄存器库允许模式

图 1.9 PSW 分布图

EMB	0	寄存器区禁止模式
	1	寄存器区允许模式

如果在直接寻址中清除该标记,则不管 SMB 的值,数据存储器的寻址范围确定为数据存储器 BANK0 的地址 000H—07FH 和 BANK15 的地址 0F80H—0FFFH。如果该标记设置为 1,数据存储器的寻址范围扩大,而且通过 SMB 的值可寻址到整个数据存储器的空间。通过 SMBn 指令可选 3 个数据存储器库其中之一为工作数据存储器(参考寻址模式)。

对于 ORG 和 VENT n (n=0—5) 指令,参考交叉汇编指南。

(例) 开始地址分配方法

```

ORG      0000H          ; ROM address assign
VENT0    1, 0, RESET    ; EMB←1, ERB←0, branch RESET
VENT1    0, 1, INTB      ; EMB←0, ERB←1, branch INTB
VENT2    0, 1, INTO      ; EMB←0, ERB←1, branch INTO
VENT3    0, 1, INT1      ; EMB←0, ERB←1, branch INT1
VENT4    0, 1, INTP0     ; EMB←0, ERB←1, branch INTP0
VENT5    6, 1, INTT      ; EMB←0, ERB←1, branch INTT
          :
RESET    BITR EMB

```

2. 寄存器库使能标记 (ERB)

ERB 是一个选择寄存器库的 1—比特标记,而且映射于 RAM 比特地址 FB2H.0。当该比特设置为 1 时,通过 SRB n 指令可选中 4 个寄存器库中的一个;当该比特清零时,不管 SRB 的值选中 BANK0。

ERB 由 RESET 向量地址和中断向量地址控制,而且也能由 BITR 和 BITS 指令控制。

ERB	0	寄存器区禁止模式
	1	寄存器区允许模式

3. 跳状态 0, 1, 2 (SC0, SC1, SC2)

这些标记表示程序的当前跳状态,它们的内容在执行程序时自动设置和清除。

这些标记映射于 RAM 比特地址 BF1H.0, FB1H.1 和 FB1H.2,而且不能处理之。

4. 中断状态标记 0, 1 (IS0, IS1)

这些标记表示当前中断的状态,不管中断什么时候产生,它们的内容自动加一,而且通过执行 IRET 指令减一,因此他们可通过 1—比特 RAM 控制指令改变,一个即将执行的中断处理的状态能够通过程序改变,如表 1-1。所以,允许多重中断。注意:在 ISx 改变前使用 DI 指令所有中断被禁止。

当产生一个中断时,ISx 的值存于堆栈区,而返回时恢复,IS0 和 IS1 映射于 RAM 比特地址 FB0H.2 和 FB0H.3。