

第 3 章 数据存储器

目录

本章包括下列主题：

3.1	简介	3-2
3.2	数据存储器构成	3-2
3.3	数据对齐	3-5
3.4	软件堆栈	3-6
3.5	程序存储空间与数据存储空间的接口	3-6
3.6	相关应用笔记	3-7
3.7	版本历史	3-8

3.1 简介

由于是哈佛架构的器件，PIC24F 单片机具有独立的程序和数据存储空间以及总线。PIC24F 架构同时还允许在代码执行过程中从数据空间直接访问程序存储器。

3.2 数据存储器构成

3.2.1 数据地址空间

PIC24F 内核具有 16 位宽的数据存储空间，可作为单个线性地址范围进行寻址。使用两个地址发生单元（Address Generation Unit，AGU）分别对数据空间执行读写操作。数据存储空间映射如图 3-1 所示。

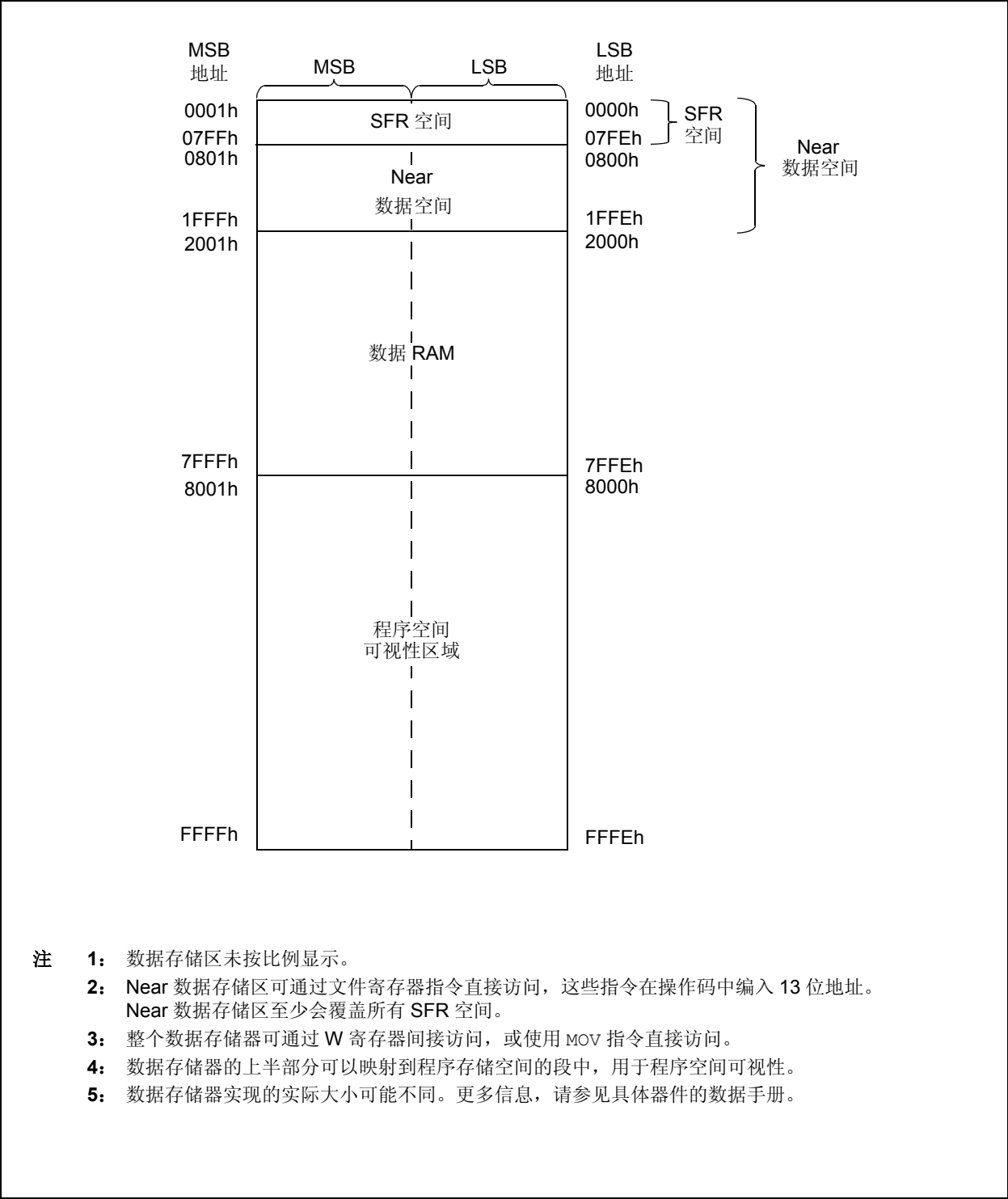
数据存储空间中的所有有效地址（Effective Address，EA）均为 16 位宽，并且指向数据空间内的字节。这使得数据空间的地址范围为 64 KB 或 32 K 字。数据存储空间的低 32 KB（即当 EA<15> = 0 时）用作实现的存储单元地址，而高半部分（EA<15> = 1）则保留为程序空间可视性（Program Space Visibility，PSV）区域。关于 PSV 的详细信息，请参见第 4.4 节“来自数据空间的程序空间可视性”。

注 1：关于具体器件中的数据存储器的实际实现，请参见具体器件的数据手册。
2：如果 EA 指向了器件中已物理实现的数据存储器区域以外的存储单元，则将返回一个全零的字或字节。

3.2.2 数据空间宽度

数据存储空间组织为可字节寻址的 16 位宽的块。在数据存储器和寄存器中的数据是以 16 位字为单位对齐的，但所有数据空间 EA 都将解析为字节。每个字的低字节（Least Significant Byte，LSB）部分具有偶地址，而高字节（Most Significant Byte，MSB）部分则具有奇地址。

图 3-1: PIC24F 器件的数据存储空间映射



3.2.3 Near 数据存储区

在 0x0000 和 0x1FFF 之间的 8 KB 地址空间被称为 near 数据存储区。Near 数据存储区可通过所有文件寄存器指令中的 13 位绝对地址域直接寻址。

Near 数据存储区也可通过所有间接寻址模式进行寻址，此时可以通过任意的 16 位工作寄存器指向数据存储区中的地址。超出 0x1FFF 的数据存储区只能通过间接寻址模式寻址。

Near 数据区域包含的存储区取决于为每个 PIC24F 系列器件型号实现的数据存储空间量。Near 数据区域至少会包含所有 SFR。更多详细信息，请参见图 3-1。

3.2.4 SFR 空间

Near 数据空间的前 2 KB 存储单元（从 0000h 到 07FFh）主要被特殊功能寄存器（Special Function Register，SFR）占用。PIC24F 内核和外设模块使用这些寄存器来控制器件的工作。

SFR 分布在受其控制的模块中，通常一个模块会使用一组 SFR。大部分 SFR 空间包含未用的地址，它们读为 0。说明实际实现的 SFR 所在地址的 SFR 空间的框图如表 3-1 所示。每个实现的区域代表一个 32 字节区域，其中至少有一个地址实现为一个 SFR。

表 3-1: SFR 数据空间的实现区域⁽¹⁾

SFR 空间地址								
	xx00	xx20	xx40	xx60	xx80	xxA0	xxC0	xxE0
000h	内核			ICN	中断			—
100h	定时器		捕捉	—	比较	—	—	—
200h	I ² C™	UART	SPI		—	—	I/O	
300h	A/D		—	—	—	—	—	—
400h	—	—	—	—	—	—	—	—
500h	—	—	—	—	—	—	—	—
600h	PMP	RTC/ 比较器	CRC	—	—	—	I/O	
700h	—	—	系统	NVM/PMD	—	—	—	—

图注： — = 在该存储块中未实现任何 SFR
注 1: 关于实际的寄存器实现，请参见具体器件的数据手册。

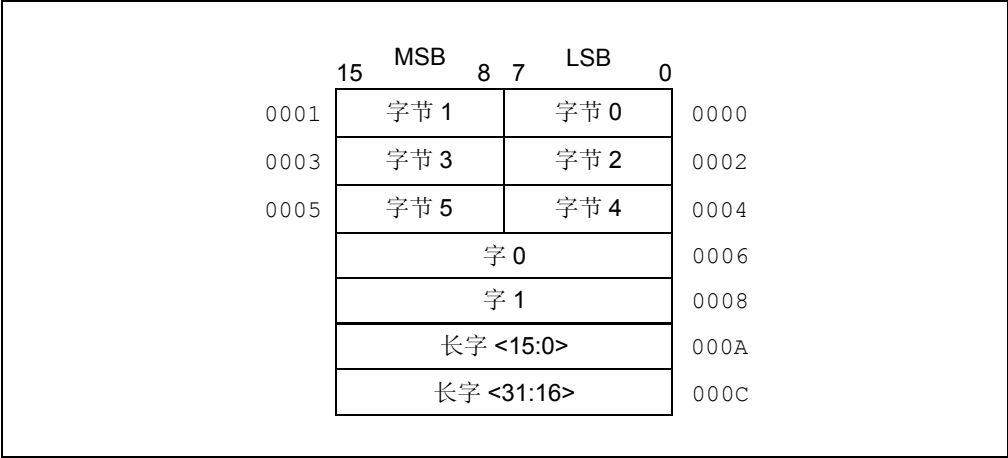
3.3 数据对齐

为维持与 PIC® 器件的后向兼容性和提高数据存储空间的使用效率，PIC24F 指令集同时支持字和字节操作。字节访问会在内部对按字对齐的存储空间的所有有效地址进行计算调整。

在字操作期间，16 位数据地址的 LSb 会被忽略。字数据按小尾数格式对齐，即低字节（LSB）位于偶地址（LSb = 0），高字节（MSB）位于奇地址（LSb = 1）。

对于字节操作，数据地址的 LSb 用于选择要访问的字节。图 3-2 给出了字操作和字节操作的数据对齐方式。

图 3-2: 数据对齐



数据字节读操作将读取包含字节的整个字，使用任何 EA 的 LSb 来确定要选取的字节。选定的字节被放在数据路径的 LSB 处。这就是说，数据存储器和寄存器被组织为两个并行的字节宽的实体，它们共享（字）地址译码，但写入线独立。数据字节写操作只写入阵列或寄存器中与字节地址匹配的那一侧。

所有有效地址的计算都是根据执行的是字节访问还是字访问而自动调整。例如，对于字操作，在操作后递增地址指针时，地址将递增 2。类似地，对于字节操作，在操作后递增地址指针时，地址将递增 1。

所有字访问必须按偶地址对齐。不支持不对齐的字数据取操作，所以在混合字节和字操作时，或者从 8 位 MCU 代码转换时，必须要小心。如果试图进行不对齐的读或写操作，将产生地址错误陷阱。如果在读操作时发生错误，正在执行的指令将完成；而如果在写操作时发生错误，指令仍将执行，但不会进行写入。无论是哪种情况，都会执行陷阱，从而系统和 / 或用户能够检查地址错误发生之前的机器状态。关于中断的更多详细信息，请参见第 8 章“中断”。

所有装入 W 寄存器的字节都将装入 W 寄存器的低字节（LSB），W 寄存器的高字节（MSB）不变。提供了一条符号扩展（SE）指令，允许用户把 8 位有符号数据转换为 16 位有符号值。或者，对于 8 位无符号数据，用户可以通过在适当地址处执行一条零扩展（ZE）指令清零任何 W 寄存器的 MSB。

尽管大多数指令能够对字或字节大小的数据进行操作，但要注意的是，部分指令只对字大小的数据进行操作。

3.4 软件堆栈

对于堆栈操作，需要将 PIC24F 器件的数据存储器的一部分分配为堆栈。关于软件堆栈的更多信息，请参见第 2.3 节“软件堆栈指针”。

3.5 程序存储空间与数据存储空间的接口

PIC24F 架构采用 24 位宽的程序空间和 16 位宽的数据空间。该架构也是一种改进型哈佛结构，这意味着数据也能存放在程序空间内。要成功使用该数据，在访问数据时必须确保这两种存储空间中的信息是对齐的。

除了正常执行外，PIC24F 架构还提供了两种可在操作过程中访问程序空间的方法：

- 使用表指令访问程序空间中任意位置的各个字节或字
- 将程序空间的一部分重新映射到数据空间（程序空间可视性）

表指令允许应用程序读写程序存储器的一小块区域。这一功能对于访问需要定期更新的数据表来说非常理想。也可通过表操作访问一个程序字的所有字节。重映射方法允许应用程序访问一大块数据，但只限于读操作，它非常适合于在一个大的静态数据表中进行查找。这一方法只能访问程序存储器的低位字。

关于程序和数据存储器接口的更多详细信息，请参见第 4.3 节“访问程序存储器中的数据”。

3.6 相关应用笔记

本节列出了与手册本章内容相关的应用笔记。这些应用笔记可能并不是专为 PIC24F 器件系列而编写的，但其概念是相关的，通过适当修改即可使用，但在使用中可能会受到一定限制。当前与数据存储器相关的应用笔记有：

标题	应用笔记编号
目前没有相关的应用笔记。	

注：如需获取更多 PIC24F 系列器件的应用笔记和代码示例，请访问 Microchip 网站（www.microchip.com）。

3.7 版本历史

版本 A（2007 年 1 月）

这是本文档的初始版本。