

## 了解并行编程协议

作者: Eckart Hartmann

### 简介

ADuC8xx系列器件编程的主要方法是通过串行编程, 详细说明请参阅ADI公司网站([www.analog.com](http://www.analog.com))上的相关数据手册以及[应用笔记AN-1074](#)。

然而, 对于某些用户来说, 通过标准器件编程器对这些器件进行编程可能更为方便。ADuC8xx系列支持并行编程, 因此标准化编程器供应商会支持该系列器件。

本应用笔记将说明该并行编程协议。此协议与市场上许多独立EPROM和EEPROM器件所用的协议本质上相同, 但有一些额外考虑必须注意。

本应用笔记的内容适用于所有ADuC83x和ADuC84x器件。

## 目录

简介.....	1	擦除全部.....	8
存储器映射.....	3	写入字节.....	9
引脚配置.....	4	页编程.....	10
进入并行编程模式.....	5	读取字节.....	11
命令功能.....	6	字节写入/字节读取程序流.....	12
写入地址寄存器.....	6		
读取地址寄存器.....	7		

2012年1月—修订版0：初始版

# 存储器映射

在并行编程模式下，存储器的各区域映射到可用可寻址空间的各部分。图1所示为小存储器器件ADuC834/ADuC836的映射，图2所示为大存储器器件ADuC831/ADuC832/

ADuC841/ADuC842/ADuC843/ADuC845/ADuC847/ADuC848的映射。MS位(详见“命令功能”部分所述)决定是对程序存储器寻址(MS = 1)，还是对数据存储器寻址(MS = 0)。

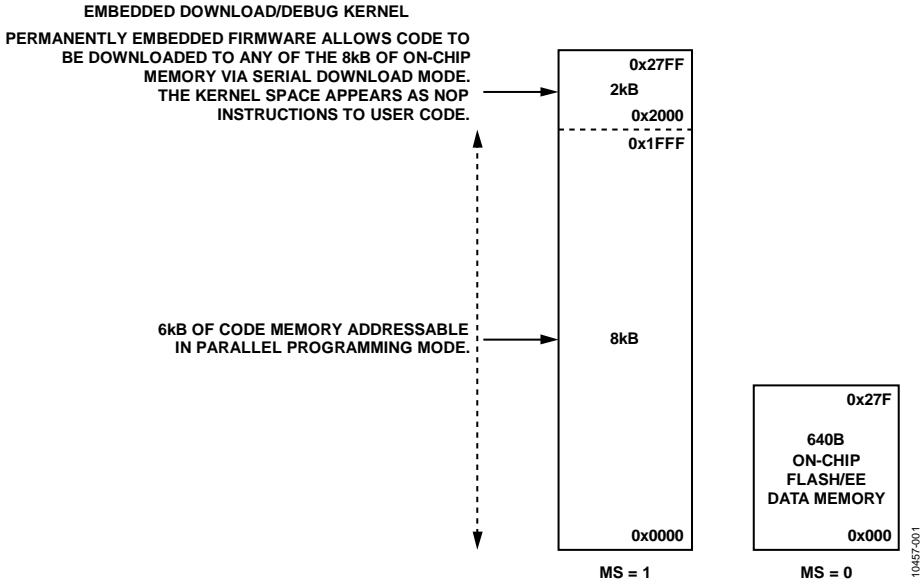


图1. 小存储器器件的并行编程存储器映射

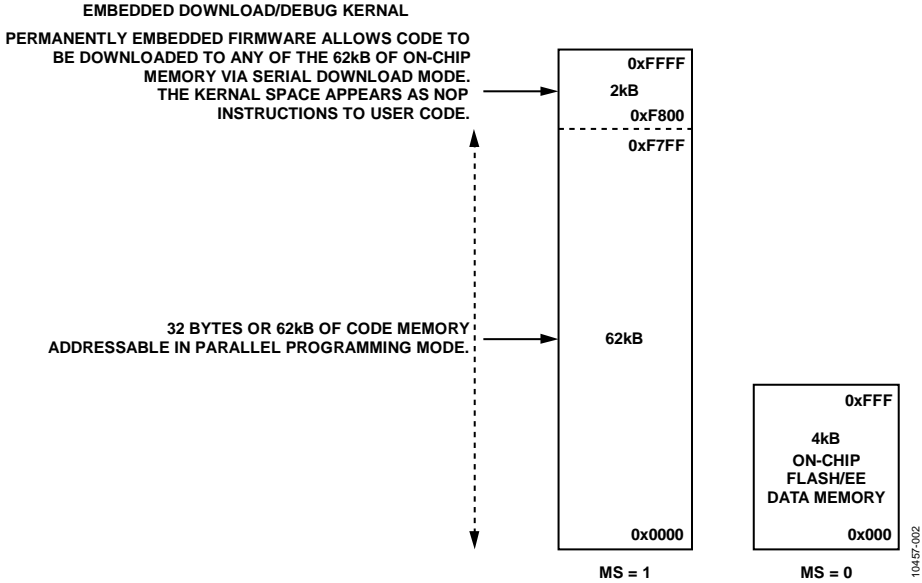


图2. 大存储器器件的并行编程存储器映射

## 引脚配置

并行编程是利用图3所示信号向器件发送一个命令序列而实现的。先是一个特殊的输入序列，然后是“命令功能”部分所述的不同命令。

- 端口3是用于写入和读取字节的8位双向数据总线。
- P1.1至P1.4是用于指定擦除、编程和读取操作的4位命令输入。
- P1.5至P1.7为并行编程提供时序。
- P1.0是低电平有效使能输入，用于选通P1.1至P1.4上的命令。
- $\overline{EA}$ 用于进入并行编程模式。
- 为了进入并行编程模式，PSEN必须通过图中所示晶体管基极的1 k $\Omega$ 电阻接地。

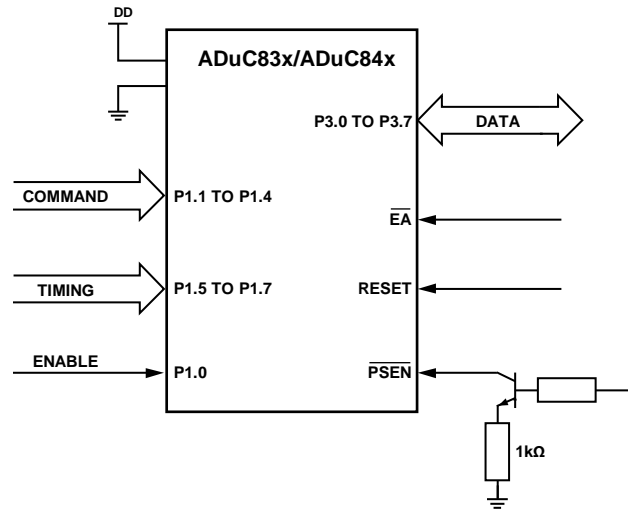


图3. 用于并行编程的引脚配置

10457-003

## 进入并行编程模式

为了进入并行编程模式，P3.1至P3.4须设置为0b1101。

进入并行模式时，为确保将一个安全命令放在端口1上，请使用0xCD命令，因为它对应于读取地址寄存器命令。

PSEN必须通过一个1 kΩ电阻驱动，如图3所示。

关于电源，应注意以下几点：

- 必须将接地引脚(DGND和AGND)视为一个节点。
- 必须将 $V_{DD}$ 引脚( $AV_{DD}$ 和 $DV_{DD}$ )视为一个节点。
- 施加于任何引脚的电压不得大于 $V_{DD}$ 或小于地。
- 在整个编程过程中， $V_{DD}$ 必须保持稳定并且处于额定范围内。

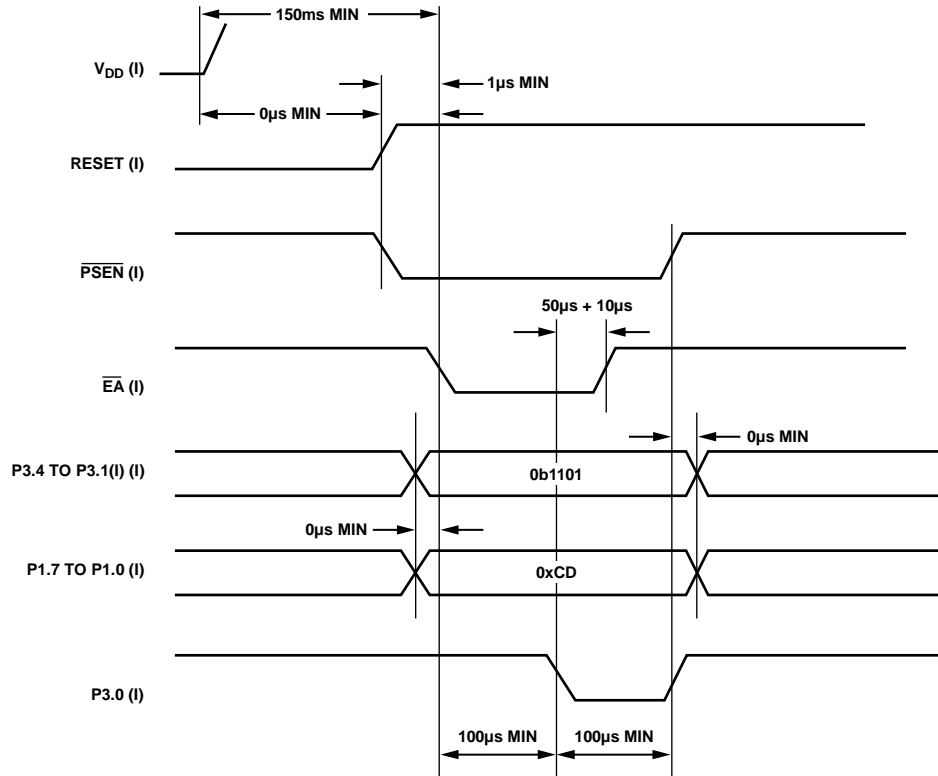


图4. 用于并行编程的进入序列

10457-004

## 命令功能

表1列出了执行各种并行编程功能的命令。

表1. 并行编程命令

P1.4	P1.3	P1.2	P1.1	功能
0	0	0	0	擦除全部(代码/数据加安全位)
MS	0	1	0	写入字节
MS	0	1	1	读取字节
0	1	1	0	读取地址寄存器
1	1	1	0	写入地址寄存器
1	1	1	1	默认上拉, 无操作

在写入字节功能和读取字节功能中, 表1中的MS位用于选择程序存储器(P1.4 = 1)或数据存储器(P1.4 = 0)。

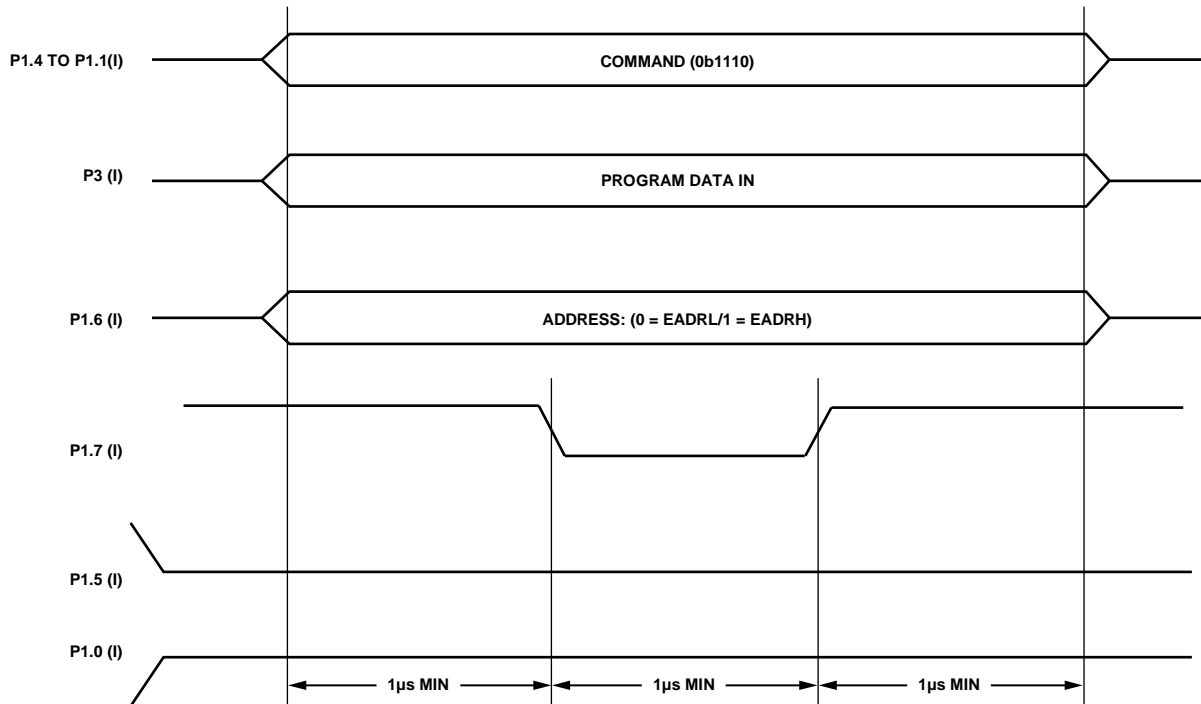
## 写入地址寄存器

EADRH和EADRL是用于微控制器产品的地址寄存器。为了写入地址进行字节编程, 请使用图5所示的命令和时序序列。

表2. 写入地址寄存器命令密钥

P1.4	P1.3	P1.2	P1.1	功能
1	1	1	0	写入地址寄存器

注意, 完成字节或页编程命令后, EADRL会自动递增, 因此对于后续的顺序编程命令, 无需手动递增EADRL。



- NOTES
1. IF P1.6 = 1, THEN EADRH IS SELECTED.
  2. IF P1.6 = 0, THEN EADRL IS SELECTED.

图5. 写入地址寄存器

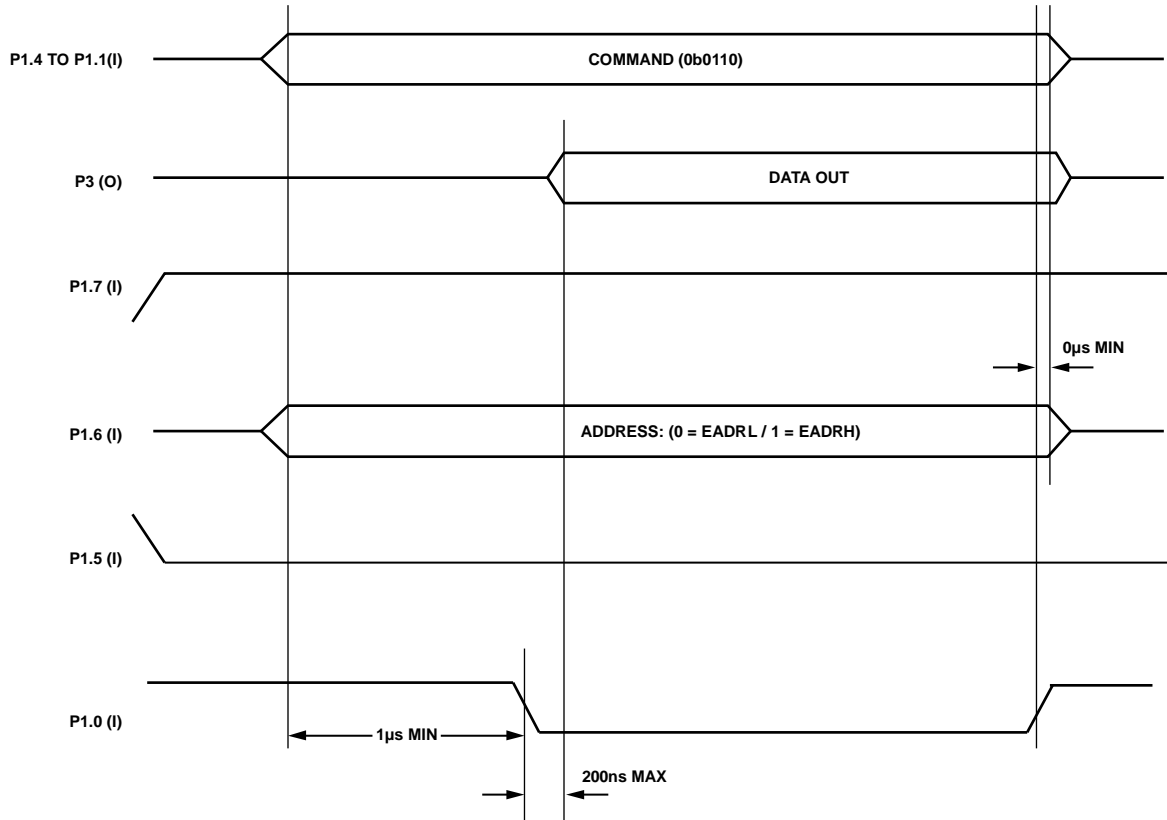
10457-005

**读取地址寄存器**

EADRH和EADRL是用于微控制器产品的地址寄存器。为了读取当前地址，请使用图6所示的命令和时序序列。

**表3. 读取地址寄存器命令密钥**

P1.4	P1.3	P1.2	P1.1	功能
0	1	1	0	读取地址寄存器



- NOTES**
1. IF P1.6 = 1, THEN EADRH IS SELECTED.
  2. IF P1.6 = 0, THEN EADRL IS SELECTED.

图6. 读取地址寄存器

10457-006

# AN-1139

## 擦除全部

为了擦除NV数据和程序闪存，请使用图7所示的命令和时序序列。

表4. 擦除全部命令密钥

P1.4	P1.3	P1.2	P1.1	功能
0	0	0	0	擦除全部(代码/数据加安全位)

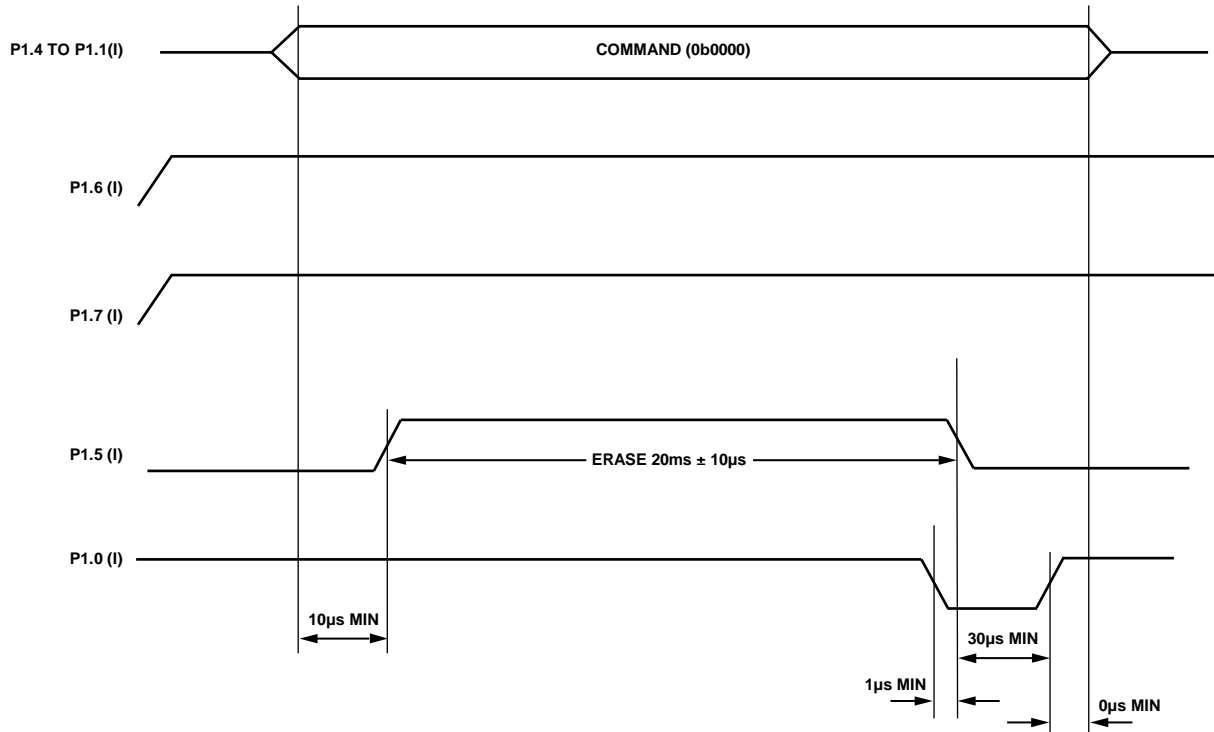


图7. 擦除全部

10457-007



**写入字节**

为了在EADRH和EADRL(已通过写入地址寄存器操作编程, 见图5)所确定的位置写入一个字节, 请使用图8所示的命令和时序序列。

**表5. 写入字节命令密钥**

P1.4	P1.3	P1.2	P1.1	功能
MS	0	1	0	写入字节

关于MS位的说明, 请参阅“命令功能”部分和表1。

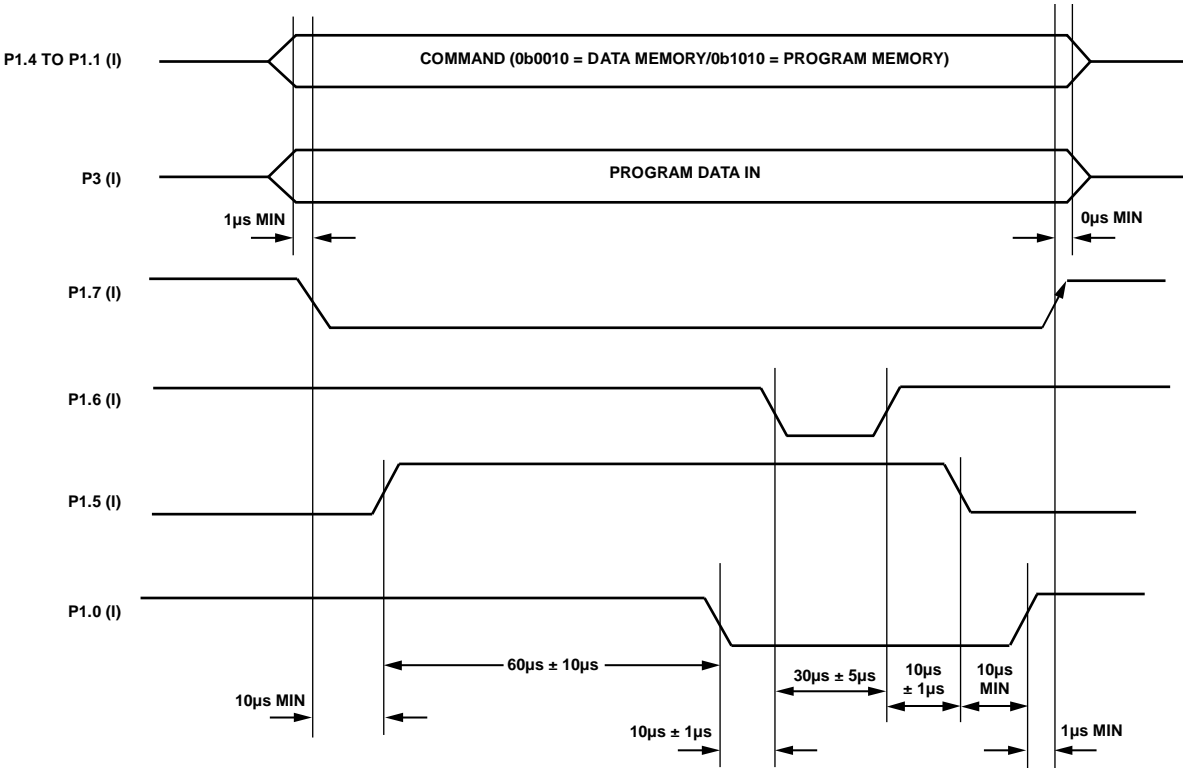


图8. 写入字节

10457-008

## 页编程

除了一次写入一个字节外，还可以使用页编程功能，该功能可以按照图9所示的时序图选择。注意，其命令字节与字节编程功能相同。事实上，页编程功能就是在严格的时序限制内连续快速写入一系列单字节。如果因为某种原因无法满足所有时序要求，请使用字节编程。

**表6. 页编程命令密钥**

P1.4	P1.3	P1.2	P1.1	功能
MS	0	1	0	写入字节

关于MS位的说明，请参阅“命令功能”部分和表1。

一页程序存储器的大小为32字节，一页数据存储器的大小仅两字节。对于程序存储器，一页的第一个地址结束于0b00000，最后一个地址结束于0b11111。对于数据存储器，一页的第一个地址结束于0，最后一个地址结束于1。详见表7和表8所示。页编程要求页中的所有地址在一个序列内依次写入，从地址0开始，结束于地址31(数据存储器则是地址1)。

**表7. 程序存储器的一页**

地址	值
地址0	XXXXXXXXXX00000
地址1	XXXXXXXXXX00001
地址2	XXXXXXXXXX00010
...	...
地址31	XXXXXXXXXX11111

**表8. 数据存储器的一页<sup>1</sup>**

地址	值
地址0	XXXXXXXXXXXXXXXXX0
地址1	XXXXXXXXXXXXXXXXX1

<sup>1</sup> 在并行编程模式下，一页数据存储器为2字节，但在用户模式下，每页为4字节。

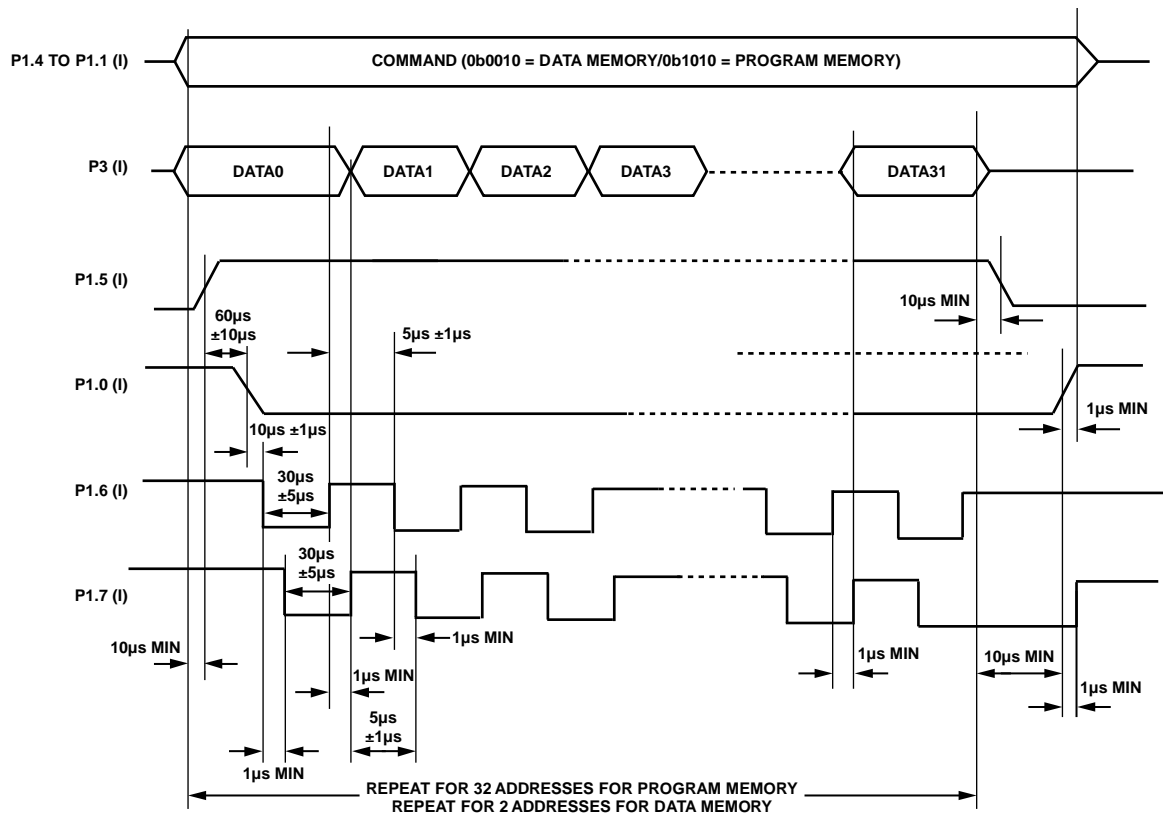


图9. 页编程

10457-009

读取字节

为了读取地址(EADRH:EADRL)处的字节，请使用图10所示的命令和时序序列。

表9. 读取字节命令密钥

P1.4	P1.3	P1.2	P1.1	功能
MS	0	1	1	读取字节

关于MS位的说明，请参阅“命令功能”部分和表1。

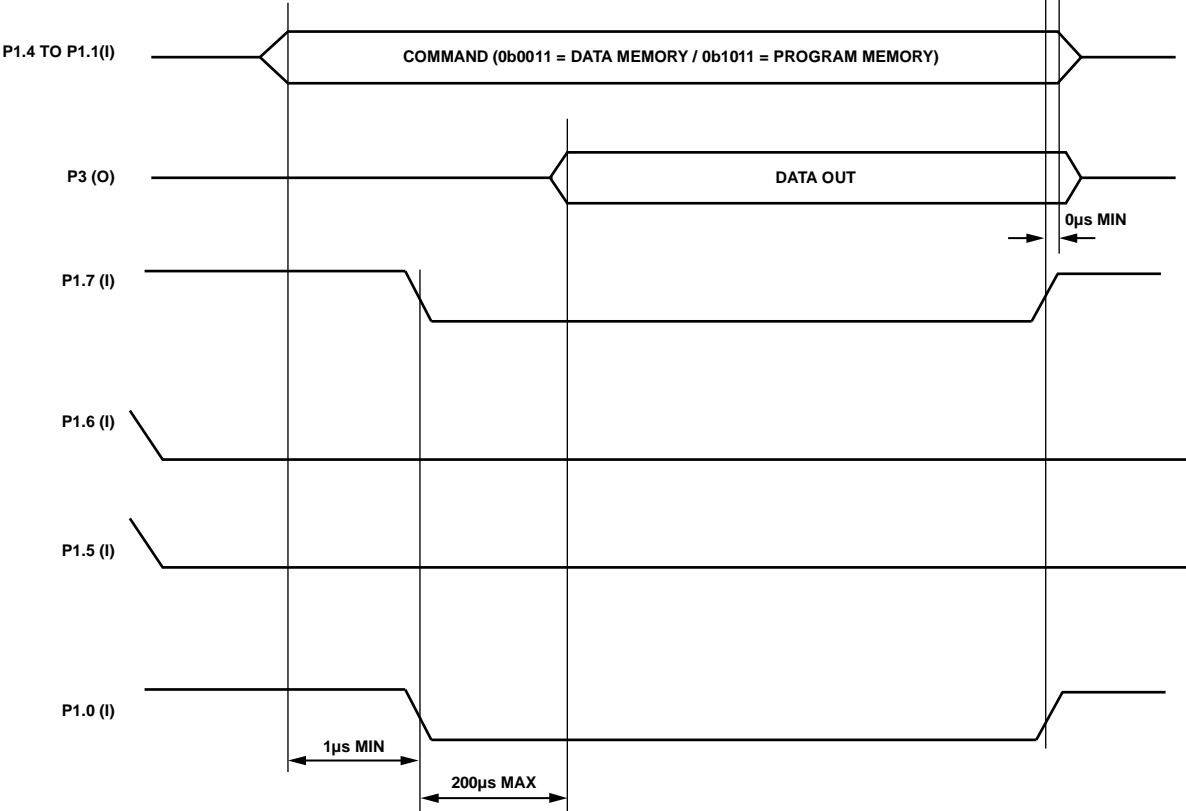


图10. 读取字节

10457-010

## 字节写入/字节读取程序流

图11所示的序列假设器件已被完全擦除。如果字节位置不是处于已擦除状态，可能无法正确编程。

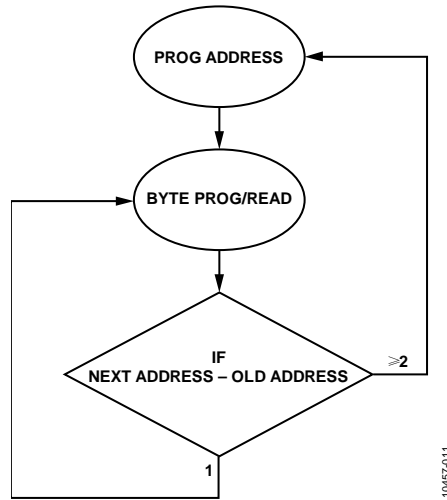


图11. 字节写入/读取序列