

John Griffith

HPA - Supervisors and Sequencers

摘要

UCD9081 的运行由器件内的可编程 (闪存) 存储器配置。可通过标准 I2C 总线接口使用命令协议对存储器进行写入和读取。本文档提供了使用配置数据对 UCD9081 进行编程所需的硬件和软件详细信息。

内容

1 引言.....	2
2 硬件.....	2
2.1 封装：RHB (S-PQFP-N32)、32 引脚塑料四方扁平封装.....	2
2.2 硬件和引脚分配.....	2
2.3 详细引脚说明.....	2
3 软件.....	3
3.1 数据文件格式.....	3
3.2 I2C 事务.....	3
3.3 器件版本.....	3
3.4 Checksum.....	3
3.5 示例配置数据文件.....	4
3.6 I2C 写入和读取事务格式.....	4
3.7 伪 I2C 写入和读取事务.....	5
4 用户配置.....	17
4.1 配置参数存储器映射.....	17
4.2 配置参数详细信息.....	18
5 其他注意事项.....	29
5.1 嵌入式应用.....	29
5.2 时序.....	29
6 参考文献.....	33
7 修订历史记录.....	33

商标

所有商标均为其各自所有者的财产。

1 引言

通过创建编程或配置参数文件，UCD9081 可针对每个独特的用户应用程序进行定制。创建配置参数文件（*.par 或 *.hex 格式）的优选高效方法是使用 TI 网站上提供的 UCD9081 评估模块 (EVM) 图形用户界面 (GUI)。本文档为 [UCD9081 数据表](#) 中包含的内容提供补充信息。

2 硬件

2.1 封装：RHB (S-PQFP-N32)、32 引脚塑料四方扁平封装

2.2 硬件和引脚分配

引脚编号	引脚名称	连接或说明
1	V _{SS}	接地
2	NC	NC
3	XIN	V _{CC}
4、17、20、31	NC ⁽¹⁾	V _{SS}
5	RST	器件复位
6	MON1	NC
7	MON2	NC
8	MON3	NC
9	MON6	NC
10	EN4	NC
11	EN3	NC
12	EN5	NC
13	EN6	NC
14	EN7	NC
15	MON7	NC
16	MON8	NC
18	MON4	NC
19	MON5	NC
21	SDA	I ² C 数据
22	SCL	I ² C 时钟
23	EN1	NC
24	EN2	NC
25	EN8/ADDR1/GPO1	V _{SS}
26	ADDR2/GPO2	V _{SS}
27	ADDR3/GPO3	V _{SS}
28	ADDR4/GPO4	V _{SS}
29	TEST	V _{SS}
30	V _{CC}	3.3V
32	ROSC	100K 至 V _{CC} ，或 1.75V

(1) 无连接

2.3 详细引脚说明

2.3.1 RST

器件复位输入：应用最小脉冲宽度为 2μs 的低电平有效信号来复位 UCD9081。在 UCD9081 处理 I2C 总线上的命令之前，对 **RST** 取反会产生延迟。有关更多详细信息，请参阅节 5.2.1。

2.3.2 SDA

I2C 串行数据输入/输出：SDA 符合 I2C 从器件的规范。在这个引脚上需要一个外部上拉电阻器。

2.3.3 SCL

I2C 串行时钟输入/输出：SCL 符合 I2C 从器件的规范。在这个引脚上需要一个外部上拉电阻器。UCD9081 可随时保持或延展 SCL（时钟延展）。在擦除期间，UCD9081 使 SCL 延展更长时间。有关更多细节信息，请参见 [节 5.2.2](#)。SCL 的最小频率为 10kHz，最大频率为 100kHz。

2.3.4 ADDR_x

器件地址输入：在对 $\overline{\text{RST}}$ 取负后不久，UCD9081 会对 ADDR1-ADDR4 进行采样，并为 I2C 总线上的器件地址定义 4 个 LSB。地址的高 3 位在 0x6 进行硬编码，提供的可寻址范围为 0x60-0x6f。使用 ADDR_x 输入端上的上拉或下拉电阻器来选择这些地址。将 ADDR1-ADDR4 连接到 VSS 会将器件定位在 I2C 地址 0x60；将 ADDR1-ADDR4 连接到 VCC 会将器件定位在 I2C 地址 0x6F。对 ADDR_x 引脚进行采样后，它们将充当 EN/GPIO，并驱动到器件配置定义的默认非活动状态。

2.3.5 RO_{SC}

振荡器输入：该引脚控制器件的运行速度。建议使用 100K 电阻上拉至 V_{cc}，但也可以在该引脚上施加 1.75V 电压。

3 软件

UCD9081 EVM GUI 为创建配置参数文件提供了理想的开发机制。可使用两个用户指南：[使用 UCD9081 进行高级时序控制和监测](#)和 [UCD9081 电源序列发生器和监测器 EVM](#)，其中包含 UCD9081 GUI 的安装和用户说明。

创建最终配置参数文件后，请按照 [UCD9081 数据表](#)中概述（并在本文档中强调）的过程仅配置 [节 3.1](#) 中描述的存储器区域。

备注

注：仅将 UCD9081EVM GUI 用于 UCD9081 器件。请勿将 UCD9080EVM GUI 用于 UCD9081 器件，因为这会损坏生成的配置参数文件。

3.1 数据文件格式

配置数据以标准 Intel 格式提供。从地址 0x1080 开始，对 128 字节数据进行编程，从地址 0xE000 开始，对 512 字节数据进行编程。[节 3.5](#) 显示了示例配置数据文件。

3.2 I2C 事务

使用配置数据对器件进行编程需要 I2C 写入事务。从器件读取配置数据需要 I2C 写入事务处理和 I2C 读取事务。读取配置数据可用于验证写入数据操作后是否正确编程了数据。[节 3.6](#) 显示了 I2C 写入和读取事务的格式。[节 3.7](#) 介绍了写入和读取 [节 3.5](#) 中显示的示例配置数据所需的一组伪 I2C 事务。

备注

[节 3.6](#) 中显示的 I2C 写入和读取数据事务假设最大数据传输块大小为 32 字节。每个块前面都有目标地址。UCD9081 支持大小从 2 字节到 512 字节的块，为两字节（一个 16 位字）的倍数。写入数据时，必须写入从地址 0x1080 开始的所有 128 字节数据，并且必须写入从地址 0xE000 开始的所有 512 字节数据（不允许对数据区进行部分写入）。

3.3 器件版本

I2C 读取事务可用于读取寄存器 0x27；值 0x55 表示 UCD9081 器件版本为 5.5。

3.4 Checksum

UCD9081 EVM GUI 会自动计算并插入地址 0xE1FE 和 0xE1FF 处的校验和字节。在 [节 3.5](#) 所示的示例文件中，这些字节以**粗体**显示。

3.5 示例配置数据文件

3.5.1 出厂默认设置

```
:20108000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF70
:2010A000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF50
:2010C000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF30
:2010E000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF10
:20E000000000000000000000000000000000000000000000000000000000
:20E0200000000000000000000000000000000000000000000000000000E0
:20E040000000000000000000000000000000000000000000000000000000
:20E060000000000000000000000000000000000000000000000000000000
:20E080000000000000000000000000000000000000000000000000000000
:20E0A0000000000000000000000000000000000000000000000000000000
:20E0C0000000000000000000000000000000000000000000000000000000
:20E0E0000000000000000000000000000000000000000000000000000000
:20E100000000000000000000000000000000000000000000000000000000
:20E120000000000000000000000000000000000000000000000000000000
:20E140000000000000000000000000000000000000000000000000000000
:20E160000000000000000000000000000000000000000000000000000000
:20E180000000000000000000000000000000000000000000000000000000
:20E1A0000000000000000000000000000000000000000000000000000000
:20E1C0000000000000000000000000000000000000000000000000000000
:20E1E00000000000000000000000000000000000000000000000000000006E347D
```

3.5.2 EVM 默认配置

```
:20108000202020202020202020202020202020202020202020202020202050
:2010A000202020202020202020202020202020202020202020202020202030
:2010C000202020202020202020202020202020202020202020202020202010
:2010E0002020202020202020202020202020202020202020202020202020F0
:20E000000000000000000000000000000000000000000000000000000000
:20E020000000000000000000000000000000000000000000000000000000E0
:20E040000000000000000000000000000000000000000000000000000000
:20E060000000000000000000000000000000000000000000000000000000
:20E080000000000000000000000000000000000000000000000000000000
:20E0A0000000000000000000000000000000000000000000000000000000
:20E0C0000000000000000000000000000000000000000000000000000000
:20E0E0000000000000000000000000000000000000000000000000000000
:20E100000000000000000000000000000000000000000000000000000000
:20E120000000000000000000000000000000000000000000000000000000
:20E140000000000000000000000000000000000000000000000000000000
:20E160000000000000000000000000000000000000000000000000000000
:20E180000000000000000000000000000000000000000000000000000000
:20E1A0000000000000000000000000000000000000000000000000000000
:20E1C0000000000000000000000000000000000000000000000000000000
:20E1E0000000000000000000000000000000000000000000000000000000003CE3
```

3.6 I2C 写入和读取事务格式

3.6.1 I2C 写入事务

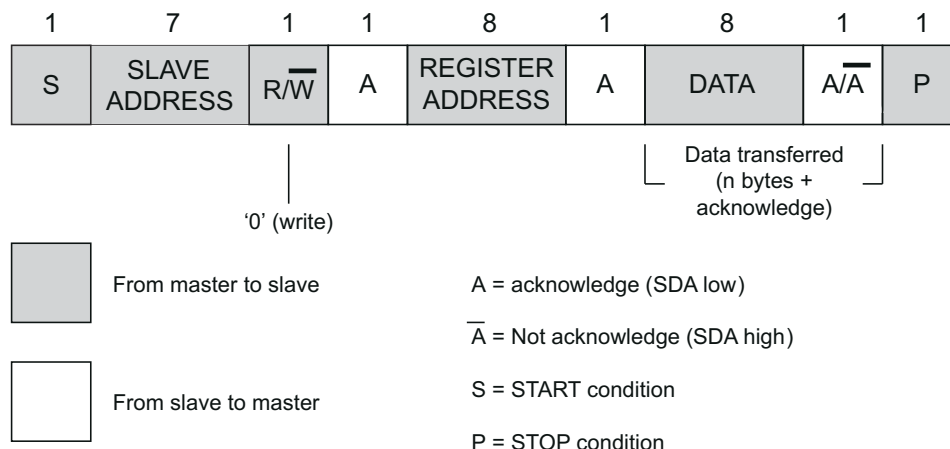


图 3-1. 写入

3.6.2 I2C 读取事务

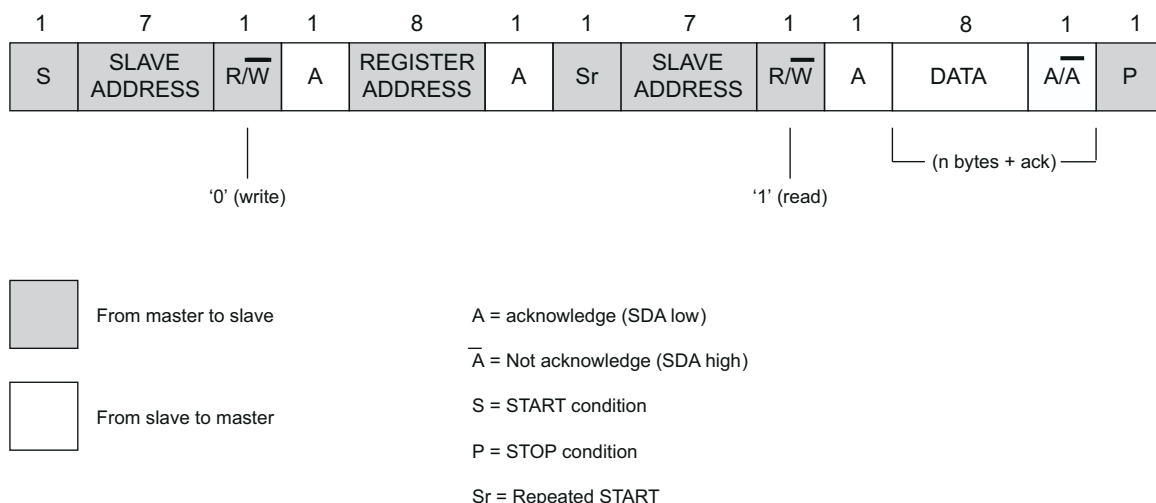


图 3-2. 读取

3.7 伪 I2C 写入和读取事务

节 3.7.1 和节 3.7.2 展示了使用 UCD9081EVM GUI (“保存 I2C 事务” 功能) 生成的伪 I2C 事务代码。请注意，在以下伪 I2C 事务中，指定了数据长度。该值不是 I2C 事务的一部分；而是在主器件内使用其值来计算传输的数据。在数据传输完成时，主器件可以向从器件生成“无应答”(NACK) 以结束事务。

3.7.1 用于写入用户数据和参数的 UCD9081 I2C 事务

I2C 写入 (打开闪存)。

 器件地址：0x6F
 寄存器地址：0x2E
 数据长度：1
 数据：0x02

I2C 写入 (基址：0x1080)

 器件地址：0x6F
 寄存器地址：0x30
 数据长度：2
 数据：0x80 0x10

I2C 写入 (解锁和擦除闪存)

----- 器件地址：0x6F
 寄存器地址：0x32
 数据长度：2
 数据：0xDC 0xBA

I2C 写入 (数据地址：0x1080)

 器件地址：0x6F
 寄存器地址：0x30
 数据长度：2
 数据：0x80 0x10

I2C 写入 (数据)

 器件地址：0x6F

寄存器地址：0x32

数据长度：32

数据：0x55 0x73 0x65 0x72 0x20 0x64 0x61 0x74 0x61 0x20 0x66 0x6F 0x72 0x20 0x45 0x56 0x4D 0x20 0x63
0x6F 0x6E 0x66 0x69 0x67 0x75 0x72 0x61 0x74 0x69 0x6F 0x6E 0x00

I2C 写入 (数据地址：0x10A0)

器件地址：0x6F

寄存器地址：0x30

数据长度：2

数据：0xA0 0x10

I2C 写入 (数据)

器件地址：0x6F

数据长度：32

数据：0x00
0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00

I2C 写入 (数据地址：0x10C0)

器件地址：0x6F

寄存器地址：0x30

数据长度：2 数据：0xC0 0x10

I2C 写入 (数据)

器件地址：0x6F

寄存器地址：0x32

数据长度：32 数据：0x00
0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00

I2C 写入 (数据地址：0x10E0)

器件地址：0x6F

寄存器地址：0x30

数据长度：2

数据：0xE0 0x10

I2C 写入 (数据)

器件地址：0x6F

寄存器地址：0x32

数据长度：32

数据：0x00
0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00

I2C 写入 (基址：0xE000)

器件地址：0x6F

寄存器地址：0x30

数据长度：2

数据：0x00 0xE0

I2C 写入 (解锁和擦除闪存)

器件地址：0x6F

寄存器地址：0x32

数据长度：2

数据：0xDC 0xBA

I2C 写入 (数据地址：0xE000)

器件地址：0x6F

寄存器地址：0x30

数据长度：2

数据：0x00 0xE0

I2C 写入 (数据)

器件地址：0x6F

寄存器地址：0x32

数据长度：32

数据：0x00 0x00

I2C 写入 (数据地址：0xE020)

器件地址：0x6F

寄存器地址：0x30

数据长度：2

数据：0x20 0xE0

I2C 写入 (数据)

器件地址：0x6F

寄存器地址：0x32

数据长度：32

数据：0x00 0x00

I2C 写入 (数据地址：0xE040)

器件地址：0x6F

寄存器地址：0x30

数据长度：2

数据：0x40 0xE0

I2C 写入 (数据)

器件地址：0x6F

寄存器地址：0x32

数据长度：32

数据：0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x0F 0xFF 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0xFF 0x00 0x00 0x00 0x00 0xC0 0x02

I2C 写入 (数据地址：0xE060)

器件地址：0x6F

寄存器地址：0x30

数据长度：2

数据：0x60 0xE0

I2C 写入 (数据)

```

-----
器件地址：0x6F
寄存器地址：0x32
数据长度：32
数据：0x00 0x00 0x00 0x0F 0x00 0x02 0x00 0x02 0xFF 0x00 0x00 0x58 0x00 0x00 0x00 0x00 0x00 0x00 0x00
0x20 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0xA8 0x00 0x10

I2C 写入 ( 数据地址：0xE080 )
-----
器件地址：0x6F
寄存器地址：0x30
数据长度：2
数据：0x80 0xE0

I2C 写入 ( 数据 )
-----
器件地址：0x6F
寄存器地址：0x32
数据长度：32
数据：0x50 0x51 0x52 0x53 0x54 0x55 0x56 0x57 0x00 0x09 0x0A 0x0B 0x00 0x00 0x00 0x00 0x00 0x00 0x00
0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00

I2C 写入 ( 数据地址：0xE0A0 )
-----
器件地址：0x6F
寄存器地址：0x30
数据长度：2
数据：0xA0 0xE0

I2C 写入 ( 数据 )
-----
器件地址：0x6F
寄存器地址：0x32
数据长度：32
数据：0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0xFF 0x7F 0xFF
0x7F 0xFF 0x7F 0xFF 0x7F 0xFF 0x7F 0xFF 0x7F 0xFF 0x7F 0xFF 0x7F

I2C 写入 ( 数据地址：0xE0C0 )
-----
器件地址：0x6F
寄存器地址：0x30
数据长度：2
数据：0xC0 0xE0

I2C 写入 ( 数据 )
-----
器件地址：0x6F
寄存器地址：0x32
数据长度：32
数据：0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00

I2C 写入 ( 数据地址：0xE0E0 )
-----
器件地址：0x6F
寄存器地址：0x30
数据长度：2

```


数据 : 0xE0 0xE0

I2C 写入 (数据)

器件地址 : 0x6F

寄存器地址 : 0x32

数据长度 : 32

数据 : 0x00

I2C 写入 (数据地址 : 0xE100)

器件地址 : 0x6F

寄存器地址 : 0x30

数据长度 : 2

数据 : 0x00 0xE1

I2C 写入 (数据)

器件地址 : 0x6F

寄存器地址 : 0x32

数据长度 : 32

数据 : 0x00

I2C 写入 (数据地址 : 0xE120)

器件地址 : 0x6F

寄存器地址 : 0x30

数据长度 : 2

数据 : 0x20 0xE1

I2C 写入 (数据)

器件地址 : 0x6F

寄存器地址 : 0x32

数据长度 : 32

数据 : 0x00 0x04 0x00 0x04 0x00 0x04 0x00 0x04 0x00 0x04 0x00 0x04 0x00 0x04 0x00 0x04 0x00 0x04 0xA0 0x0F 0xA0 0x0F 0xA0 0x0F 0xA0 0x0F 0xA0 0x0F 0xA0 0x0F 0xA0 0x0F

I2C 写入 (数据地址 : 0xE140)

器件地址 : 0x6F

寄存器地址 : 0x30

数据长度 : 2

数据 : 0x40 0xE1

I2C 写入 (数据)

器件地址 : 0x6F

寄存器地址 : 0x32

数据长度 : 32

数据 : 0x10 0x00 0x10 0x00 0x10 0x00 0x10 0x00 0x10 0x00 0x10 0x00 0x10 0x00 0x10 0x00 0x10 0x00 0x10 0x00 0x10 0x00 0x10 0x00 0x10 0x00 0x10 0x00 0x10 0x00

I2C 写入 (数据地址 : 0xE160)

器件地址：0x6F
寄存器地址：0x30
数据长度：2
数据：0x60 0xE1

I2C 写入 (数据)

器件地址：0x6F
寄存器地址：0x32
数据长度：32
数据：0x00 0x00 0x00 0xC0 0x00 0xC0 0x00 0xC0 0x04 0x20 0x08 0x20 0x04 0x18 0x02 0x18 0x08 0x18 0x10
0x18 0x20 0x18 0x10 0x20 0x00 0x20 0x20 0x20 0x40 0x20 0x80 0x20

I2C 写入 (数据地址：0xE180)

器件地址：0x6F
寄存器地址：0x30
数据长度：2
数据：0x80 0xE1

I2C 写入 (数据)

器件地址：0x6F
寄存器地址：0x32
数据长度：32
数据：0x00 0x00 0x00 0x04 0xD4 0x02 0xF2 0x08 0x10 0x01 0x05 0xC0 0x55 0x00 0x05 0x00 0x05 0xFF 0x00
0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00

I2C 写入 (数据地址：0xE1A0)

器件地址：0x6F
寄存器地址：0x30
数据长度：2
数据：0xA0 0xE1

I2C 写入 (数据)

器件地址：0x6F
寄存器地址：0x32
数据长度：32
数据：0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00

I2C 写入 (数据地址：0xE1C0)

器件地址：0x6F
寄存器地址：0x30
数据长度：2
数据：0xC0 0xE1

I2C 写入 (数据)

器件地址：0x6F
寄存器地址：0x32
数据长度：32
数据：0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00

■■■■■■■■■■

数据：0xE0 0xE1

数据：0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x3C

.....

数据：0x00

3.7.2 用于读取用户数据和参数的 UCD9081 I2C 事务

I2C 写入 (数据地址 : 0x1080)

器件地址 : 0x6F

寄存器地址 : 0x30

数据长度 : 2

数据 : 0x80 0x10

I2C 读取 (数据)

器件地址 : 0x6F

寄存器地址 : 0x32

数据长度 : 32

I2C 写入 (数据地址 : 0x10A0)

器件地址 : 0x6F

寄存器地址 : 0x30

数据长度 : 2

数据 : 0xA0 0x10

I2C 读取 (数据)

器件地址 : 0x6F

寄存器地址 : 0x32

数据长度 : 32

I2C 写入 (数据地址 : 0x10C0)

器件地址 : 0x6F

寄存器地址 : 0x30

数据长度 : 2

数据 : 0xC0 0x10

I2C 读取 (数据)

器件地址 : 0x6F

寄存器地址 : 0x32

数据长度 : 32

I2C 写入 (数据地址 : 0x10E0)

器件地址 : 0x6F

寄存器地址 : 0x30

数据长度 : 2

数据 : 0xE0 0x10

I2C 读取 (数据)

器件地址 : 0x6F

寄存器地址 : 0x32

数据长度 : 32

I2C 写入 (数据地址 : 0xE000)

器件地址 : 0x6F

寄存器地址 : 0x30

数据长度：2

数据：0x00 0xE0

I2C 读取 (数据)

器件地址：0x6F

寄存器地址：0x32

数据长度：32

I2C 写入 (数据地址：0xE020)

器件地址：0x6F

寄存器地址：0x30

数据长度：2

数据：0x20 0xE0

I2C 读取 (数据)

器件地址：0x6F

寄存器地址：0x32

数据长度：32

I2C 写入 (数据地址：0xE040)

器件地址：0x6F

寄存器地址：0x30

数据长度：2

数据：0x40 0xE0

I2C 读取 (数据)

器件地址：0x6F

寄存器地址：0x32

数据长度：32

I2C 写入 (数据地址：0xE060)

器件地址：0x6F

寄存器地址：0x30

数据长度：2

数据：0x60 0xE0

I2C 读取 (数据)

器件地址：0x6F

寄存器地址：0x32

数据长度：32

I2C 写入 (数据地址：0xE080)

器件地址：0x6F

寄存器地址：0x30

数据长度：2

数据：0x80 0xE0

I2C 读取 (数据)

器件地址：0x6F

寄存器地址：0x32

数据长度：32

I2C 写入 (数据地址：0xE0A0)

器件地址：0x6F

寄存器地址：0x30

数据长度：2

数据：0xA0 0xE0

I2C 读取 (数据)

器件地址：0x6F

寄存器地址：0x32

数据长度：32

I2C 写入 (数据地址：0xE0C0)

器件地址：0x6F

寄存器地址：0x30

数据长度：2

数据：0xC0 0xE0

I2C 读取 (数据)

器件地址：0x6F

寄存器地址：0x32

数据长度：32

I2C 写入 (数据地址：0xE0E0)

器件地址：0x6F

寄存器地址：0x30

数据长度：2

数据：0xE0 0xE0

I2C 读取 (数据)

器件地址：0x6F

寄存器地址：0x32

数据长度：32

I2C 写入 (数据地址：0xE100)

器件地址：0x6F

寄存器地址：0x30

数据长度：2

数据：0x00 0xE1

I2C 读取 (数据)

器件地址：0x6F

寄存器地址：0x32

数据长度：32

I2C 写入 (数据地址：0xE120)

器件地址：0x6F

寄存器地址：0x30
数据长度：2
数据：0x20 0xE1

I2C 读取 (数据)

器件地址：0x6F
寄存器地址：0x32
数据长度：32

I2C 写入 (数据地址：0xE140)

器件地址：0x6F
寄存器地址：0x30
数据长度：2
数据：0x40 0xE1

I2C 读取 (数据)

器件地址：0x6F
寄存器地址：0x32
数据长度：32

I2C 写入 (数据地址：0xE160)

----- 器件地址：0x6F

寄存器地址：0x30
数据长度：2
数据：0x60 0xE1

I2C 读取 (数据)

器件地址：0x6F
寄存器地址：0x32
数据长度：32

I2C 写入 (数据地址：0xE180)

器件地址：0x6F
寄存器地址：0x30
数据长度：2
数据：0x80 0xE1

I2C 读取 (数据)

器件地址：0x6F
寄存器地址：0x32
数据长度：32

I2C 写入 (数据地址：0xE1A0)

器件地址：0x6F
寄存器地址：0x30
数据长度：2
数据：0xA0 0xE1

I2C 读取 (数据)

器件地址：0x6F

寄存器地址：0x32

数据长度：32

I2C 写入 (数据地址：0xE1C0)

器件地址：0x6F

寄存器地址：0x30

数据长度：2

数据：0xC0 0xE1

I2C 读取 (数据)

器件地址：0x6F

寄存器地址：0x32

数据长度：32

I2C 写入 (数据地址：0xE1E0)

器件地址：0x6F

寄存器地址：0x30

数据长度：2

数据：0xE0 0xE1

I2C 读取 (数据)

器件地址：0x6F

寄存器地址：0x32

数据长度：32

4 用户配置

4.1 配置参数存储器映射

表 4-1 显示了 512 字节配置参数存储器映射。以粗体显示的用户可配置字节在节 4.2 中进行了说明；用户可配置字节的相邻组在表格中通过交替使用斜体进行区分。其他字节必须完全如表 4-1 中所示的那样。

表 4-1. 配置参数存储器映射

地址	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
E000	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00
E010	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00
E020	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00
E030	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00
E040	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0xFF	0xFF	0x00	0x00	0x00	0x00	0x00	0x00
E050	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0xFF	0x00	0x00	0x00	0x00	0x00	0xC0	0x02
E060	0x00	0x00	0x00	0x0F	0x00	0x02	0x00	0x02	0x00	0x00	0x00	0x58	0x00	0x00	0x00	0x00
E070	0x00	0x00	0x00	0x20	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0xA8	0x00	0x10
E080	0x10	0x11	0x12	0x13	0x14	0x15	0x16	0x17	0x00	0x09	0x0A	0x0B	0x00	0x01	0x02	0x03
E090	0x04	0x05	0x06	0x07	0x00	0x00	0x00	0x00	0x05	0x00	0x0A	0x00	0x0E	0x00	0x14	0x00
E0A0	0x19	0x00	0x1E	0x00	0x23	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00
E0B0	0xFF	0x7F	0xFF	0x7F	0xFF	0x7F	0xFF	0x7F	0xFF	0x7F	0xFF	0x7F	0xFF	0x7F	0xFF	0x7F
E0C0	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00
E0D0	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00
E0E0	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00
E0F0	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00
E100	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00
E110	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00
E120	0x00	0x04	0x00	0x04	0x00	0x04	0x00	0x04	0x00	0x04	0x00	0x04	0x00	0x04	0x00	0x04
E130	0xA0	0x0F	0xA0	0x0F	0xA0	0x0F	0xA0	0x0F	0xA0	0x0F	0xA0	0x0F	0xA0	0x0F	0xA0	0x0F
E140	0x10	0x00	0x10	0x00	0x10	0x00	0x10	0x00	0x10	0x00	0x10	0x00	0x10	0x00	0x10	0x00
E150	0x00	0xC0	0x00	0xC0	0x00	0xC0	0x00	0xC0	0x00	0xC0	0x00	0xC0	0x00	0xC0	0x00	0xC0
E160	0x00	0x00	0x00	0xC0	0x00	0xC0	0x00	0xC0	0x04	0x20	0x08	0x20	0x04	0x18	0x02	0x18
E170	0x08	0x18	0x10	0x18	0x20	0x18	0x10	0x20	0x00	0x20	0x20	0x20	0x40	0x20	0x80	0x20
E180	0x00	0x00	0x00	0x04	0xD4	0x02	0xF2	0x08	0x10	0x01	0x05	0xC0	0x55	0x00	0xFF	0x08
E190	0x05	0xFF	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00
E1A0	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00
E1B0	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00
E1C0	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00
E1D0	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00
E1E0	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00
E1F0	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x6E	0x34

4.2 配置参数详细信息

4.2.1 GpDir

配置参数中的 GpDir 字段用于指定 RAILn 或 GPOn 的方向 (输入或输出)。这应与所需的使能设置相对应。如果启用了 RAIL 或 GPO，则还应将其设置为输出。寄存器内容如下所示：

地址	容量	默认值	说明
0xE048	2	0xFFFF	GPOn 和 RAILn 的输入/输出参数

寄存器的格式如下：

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	GPO4	GPO3	GPO2	GPO1	RAIL8	RAIL7	RAIL6	RAIL5	RAIL4	RAIL3	RAIL2	RAIL1

RAILn 或 GPOn	含义
0	将 RAILn 或 GPOn 设置为输出 (使能)
1	将 RAILn 或 GPOn 设置为输入 (禁用)

4.2.2 NegateEnablePolarity

配置参数中的 NegateEnablePolarity 字段允许为 RAILn 和 GPOn 选择高电平有效或低电平有效的使能信号。这些值应与 **EnablePolarity** 寄存器 (R[0xE168] 至 R[0xE17E]) 的位 16 对齐 (请参阅节 4.2.13)。如果 RAIL 或 GPO 设置为使能低电平有效 (值 = 0b)，则相应的 NegateEnablePolarity 位值也应设置为 1b。寄存器内容如下所示：

地址	容量	默认值	说明
0xE04A	2	0x0000	Railn 和 GPOn 的 NegateEnable 信号参数

寄存器的格式如下：

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	GPO4	GPO3	GPO2	GPO1	RAIL8	RAIL7	RAIL6	RAIL5	RAIL4	RAIL3	RAIL2	RAIL1

RAILn 或 GPOn	含义
0	在 R[0xE168]:R[0xE17E] 位 [16] 中，RAILn 或 GPON 的使能信号设置为高电平有效 (1b)
1	在 R[0xE168]:R[0xE17E] bit[16] 中，RAILn 或 GPON 的使能信号设置为低电平有效 (0b)

4.2.3 SeqEventPending

配置参数中的 SeqEventPending 字段用于确定是否启用或禁用 RAILn 或 GPON 以进行时序控制。这些值应与 **SequenceEventParameter** 保持一致 (请参阅节 4.2.4)，如果 RAILn/GPON 未被禁用，也不应在这些参数中禁用它。

地址	容量	默认值	说明
0xE068	2	0x0000	GPOn 和 RAILn 的时序控制参数

寄存器的格式如下：

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	GPO4	GPO3	GPO2	GPO1	RAIL8	RAIL7	RAIL6	RAIL5	RAIL4	RAIL3	RAIL2	RAIL1

RAILn 或 GPOn	含义
0	为 RAILn 或 GPOn 禁用时序控制
1	为 RAILn 或 GPOn 启用时序控制

4.2.4 SequenceEventParameters

配置参数中的 SequenceEventParameters 字段用于指定每个电源轨的电源轨识别、监测状态和时序控制选项。这些寄存器的地址映射如下：

地址	容量	默认值	说明
0xE080	1	0x50	电源轨 1 识别、监测状态和时序控制选项
0xE081	1	0x51	电源轨 2 识别、监测状态和时序控制选项
0xE082	1	0x52	电源轨 3 识别、监测状态和时序控制选项
0xE083	1	0x53	电源轨 4 识别、监测状态和时序控制选项
0xE084	1	0x54	电源轨 5 识别、监测状态和时序控制选项
0xE085	1	0x55	电源轨 6 识别、监测状态和时序控制选项
0xE086	1	0x56	电源轨 7 识别、监测状态和时序控制选项
0xE087	1	0x57	电源轨 8 识别、监测状态和时序控制选项
0xE088	1	0x00	GPO1 识别、时序控制选项
0xE089	1	0x49	GPO2 识别、时序控制选项
0xE08A	1	0x4A	GPO3 识别、时序控制选项
0xE08B	1	0x4B	GPO4 识别、时序控制选项

每个寄存器的格式如下：

7	6	5	4	3	2	1	0
ENABLE	0	MON	RAIL/GPO				

RAIL

Rail #(n) – 1, RAIL = 0 through 7

GPO

GPO #(n) + 7, GPO = 8, 9, 0xA, 0xB

MON

Meaning

- 0 Do not monitor rail status (for event sequencing of GPOs)
- 1 Monitor rail status

ENABLE

Meaning

- 00 Sequence is disabled
- 01 Sequence is triggered after delay after sequence event
- 10 Sequence is triggered after parent rail achieves voltage level
- 11 Sequence is triggered after delay after parent rail achieves voltage regulation

4.2.5 SequenceEventLink

SequenceEventLink 字段允许为每个 ENx 和 GPOx 输出指定父轨（受监测的输入）。RESEQ 位（关断后的序列）允许标记使能或 GPO，以便在系统关断后对其进行时序控制（由当前序列发生器配置定义）。这些寄存器的地址映射如下：

地址	容量	默认值	说明
0xE08C	1	0x01	电源轨 1 父电源轨标识符和重新时序控制指示器
0xE08D	1	0x00	电源轨 2 父电源轨标识符和重新时序控制指示器
0xE08E	1	0x01	电源轨 3 父电源轨标识符和重新时序控制指示器
0xE08F	1	0x04	电源轨 4 父电源轨标识符和重新时序控制指示器
0xE090	1	0x01	电源轨 5 父电源轨标识符和重新时序控制指示器
0xE091	1	0x04	电源轨 6 父电源轨标识符和重新时序控制指示器
0xE092	1	0x05	电源轨 7 父电源轨标识符和重新时序控制指示器
0xE093	1	0x06	电源轨 8 父电源轨标识符和重新时序控制指示器
0xE094	1	0x00	GPO1 父电源轨标识符和重新时序控制指示器
0xE095	1	0x00	GPO2 父电源轨标识符和重新时序控制指示器
0xE096	1	0x00	GPO3 父电源轨标识符和重新时序控制指示器
0xE097	1	0x00	GPO4 父电源轨标识符和重新时序控制指示器

每个寄存器的格式如下：

7	6	5	4	3	2	1	0
0	RESEQ	0	0	PARENTAIL			

RESEQ

RESEQ	Meaning
0	Do not sequence after shutdown.
1	Sequence after shutdown.

PARENTAIL

PARENTAIL	Meaning
0x0000	Sequence is dependent on RAIL1 achieving the specified event.
0x0001	Sequence is dependent on RAIL2 achieving the specified event.
0x0010	Sequence is dependent on RAIL3 achieving the specified event.
0x0011	Sequence is dependent on RAIL4 achieving the specified event.
0x0100	Sequence is dependent on RAIL5 achieving the specified event.
0x0101	Sequence is dependent on RAIL6 achieving the specified event.
0x0110	Sequence is dependent on RAIL7 achieving the specified event.
0x0111	Sequence is dependent on RAIL8 achieving the specified event.

4.2.6 SequenceEventData

配置参数中的 SequenceEventData 字段用于指定电源轨和 GPO 时序控制和关断参数。这些寄存器的地址映射如下：

地址	容量	默认值	说明
0xE098	2	0xE005	电源轨 1 时序控制和关断参数
0xE09A	2	0xA005	电源轨 2 时序控制和关断参数
0xE09C	2	0xE032	电源轨 3 时序控制和关断参数
0xE09E	2	0xE033	电源轨 4 时序控制和关断参数
0xE0A0	2	0xE033	电源轨 5 时序控制和关断参数
0xE0A2	2	0xE035	电源轨 6 时序控制和关断参数
0xE0A4	2	0xE035	电源轨 7 时序控制和关断参数
0xE0A6	2	0x0000	电源轨 8 时序控制和关断参数
0xE0A8	2	0x0000	GPO1 时序控制和关断参数
0xE0AA	2	0x0000	GPO2 时序控制和关断参数
0xE0AC	2	0x0000	GPO3 时序控制和关断参数
0xE0AE	2	0x0000	GPO4 时序控制和关断参数

每个寄存器的格式如下。SequenceEventParameters 寄存器的 ENABLE 字段中的值用于确定 SequenceEventData 寄存器的 RAILDATA 字段中的值所表示的度量。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SEQPARAM				0	RAILDATA										

SEQPARAM	Meaning	ENABLE (SequenceEventParameters)	RAILDATA Meaning
000	Log only		
001	Sequence	01	Delay (in units of ms)
010	Reserved	10	Voltage (in units of Vref/1024 volts)
011	Reserved	11	Delay (in units of ms)
100	Reserved		
101	Retry 1 times		
110	Retry 0 times		
111	Reserved		

4.2.7 DependencyMasks

配置参数中的 **DependencyMasks** 字段定义用于电源轨和 **GPO** 关断的电源轨依赖掩码。此掩码表示在该电源轨关断时必须关断的其他电源轨和 **GPO** 集。请注意，由于仅监测电源轨，因此该表仅包含电源轨关断条目。在依赖掩码本身中，有一些位允许 **GPO** 关断。

这些寄存器的地址映射如下：

地址	容量	默认值	说明
0xE100	2	0x007F	电源轨 1 的依赖掩码
0xE102	2	0x0001	电源轨 2 的依赖掩码
0xE104	2	0x0002	电源轨 3 的依赖掩码
0xE106	2	0x0004	电源轨 4 的依赖掩码
0xE108	2	0x0008	电源轨 5 的依赖掩码
0xE10A	2	0x0010	电源轨 6 的依赖掩码
0xE10C	2	0x0020	电源轨 7 的依赖掩码
0xE10E	2	0x0040	电源轨 8 的依赖掩码

每个寄存器的格式如下：

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	GPO4	GPO3	GPO2	GPO1	RAIL8	RAIL7	RAIL6	RAIL5	RAIL4	RAIL3	RAIL2	RAIL1

RAILn or GPOn	Meaning
0	Shutdown of this rail does not shut down RAILn or GPOn.
1	Shutdown of this rail shuts down RAILn or GPOn.

4.2.8 UnderVoltageThresholds

配置参数中的 UnderVoltageThresholds 字段用于指定在监测该电源轨时使用的每个电源轨欠压阈值。这些寄存器的地址映射如下：

地址	容量	默认值	说明
0xE110	2	0x0000	电源轨 8 的欠压阈值
0xE112	2	0x0000	电源轨 7 的欠压阈值
0xE114	2	0x0000	电源轨 6 的欠压阈值
0xE116	2	0x0000	电源轨 5 的欠压阈值
0xE118	2	0x0000	电源轨 4 的欠压阈值
0xE11A	2	0x0000	电源轨 3 的欠压阈值
0xE11C	2	0x0000	电源轨 2 的欠压阈值
0xE11E	2	0x0000	电源轨 1 的欠压阈值

每个寄存器的格式如下：

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	Vraw										

电压转换取决于所配置的电压基准以及电路板上用于每个电源轨的上拉/下拉电阻。所选择的电压基准为 2.5V (内部) 或 V_{CC} (外部)。将所需的电源轨 UnderVoltageThreshold 转换为 Vraw 的公式如下：

不带外部电源轨分压器：

$$V_{raw} = \frac{1024 \times V_{RAILUV}}{V_{REF}} \quad (1)$$

带外部电源轨分压器：

$$V_{raw} = \frac{1024 \times V_{RAILUV}}{V_{REF}} \times \frac{R_{PULLDOWN}}{R_{PULLDOWN} + R_{PULLUP}} \quad (2)$$

4.2.9 过压阈值

配置参数中的 **OverVoltageThresholds** 字段用于指定在监测该电源轨时使用的每个电源轨过压阈值。这些寄存器的地址映射如下：

地址	容量	默认值	说明
0xE120	2	0x0400	电源轨 8 的过压阈值
0xE122	2	0x0400	电源轨 7 的过压阈值
0xE124	2	0x0400	电源轨 6 的过压阈值
0xE126	2	0x0400	电源轨 5 的过压阈值
0xE128	2	0x0400	电源轨 4 的过压阈值
0xE12A	2	0x0400	电源轨 3 的过压阈值
0xE12C	2	0x0400	电源轨 2 的过压阈值
0xE12E	2	0x0400	电源轨 1 的过压阈值

每个寄存器的格式如下：

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	Vraw										

电压转换取决于所配置的电压基准以及电路板上用于每个电源轨的上拉/下拉电阻。所选择的电压基准为 2.5V (内部) 或 V_{CC} (外部)。将所需的电源轨过压阈值转换为 **Vraw** 的公式如下：

不带外部电源轨分压器：

$$V_{raw} = \frac{1024 \times V_{RAILOV}}{V_{REF}} \quad (3)$$

带外部分压器：

$$V_{raw} = \frac{1024 \times V_{RAILOV}}{V_{REF}} \times \frac{R_{PULLDOWN}}{R_{PULLDOWN} + R_{PULLUP}} \quad (4)$$

4.2.10 RampTime

配置参数中的 **RampTime** 字段用于指定每个电源轨实现稳压所需的最长时间。这些寄存器的地址映射如下：

地址	容量	默认值	说明
0xE130	2	0x0FA0	电源轨 1 的最大电压斜坡时间
0xE132	2	0x0FA0	电源轨 2 的最大电压斜坡时间
0xE134	2	0x0FA0	电源轨 3 的最大电压斜坡时间
0xE136	2	0x0FA0	电源轨 4 的最大电压斜坡时间
0xE138	2	0x0FA0	电源轨 5 的最大电压斜坡时间
0xE13A	2	0x0FA0	电源轨 6 的最大电压斜坡时间
0xE13C	2	0x0FA0	电源轨 7 的最大电压斜坡时间
0xE13E	2	0x0FA0	电源轨 8 的最大电压斜坡时间

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	RAMPTIME											

$$RAMPTIME = RAILn \text{ RailTime (in units of ms).}$$

4.2.11 OutOfRegulationWidth

配置参数中的 OutOfRegulationWidth 字段用于指定在声明错误之前，允许电源轨不进行稳压调节的最长时间（干扰持续时间）。这些寄存器的地址映射如下：

地址	容量	默认值	说明
0xE140	2	0x0010	在不标记错误的情况下，电源轨 1 允许的无稳压调节持续时间
0xE142	2	0x0010	在不标记错误的情况下，电源轨 2 允许的无稳压调节持续时间
0xE144	2	0x0010	在不标记错误的情况下，电源轨 3 允许的无稳压调节持续时间
0xE146	2	0x0010	在不标记错误的情况下，电源轨 4 允许的无稳压调节持续时间
0xE148	2	0x0010	在不标记错误的情况下，电源轨 5 允许的无稳压调节持续时间
0xE14A	2	0x0010	在不标记错误的情况下，电源轨 6 允许的无稳压调节持续时间
0xE14C	2	0x0010	在不标记错误的情况下，电源轨 7 允许的无稳压调节持续时间
0xE14E	2	0x0010	在不标记错误的情况下，电源轨 8 允许的无稳压调节持续时间

此寄存器的内容如下：

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	OORW											

OORW = RAILn out-of-regulation glitch width (in units of 1/10 ms).

4.2.12 UnsequenceTime

配置参数中的 UnsequenceTime 字段用于指定每个电源轨在取消时序控制之前必须延迟的时间量。这些寄存器的地址映射如下：

地址	容量	默认值	说明
0xE150	2	0xC0FF	电源轨 1 的取消时序控制延迟
0xE152	2	0xC1FF	电源轨 2 的取消时序控制延迟
0xE154	2	0xC2FF	电源轨 3 的取消时序控制延迟
0xE156	2	0xC3FF	电源轨 4 的取消时序控制延迟
0xE158	2	0xC4FF	电源轨 5 的取消时序控制延迟
0xE15A	2	0xC5FF	电源轨 6 的取消时序控制延迟
0xE15C	2	0xC6FF	电源轨 7 的取消时序控制延迟
0xE15E	2	0xC7FF	电源轨 8 的取消时序控制延迟
0xE160	2	0x0000	GPO1 的取消时序控制延迟
0xE162	2	0xC000	GPO2 的取消时序控制延迟
0xE164	2	0xC000	GPO3 的取消时序控制延迟
0xE166	2	0xC000	GPO4 的取消时序控制延迟

此寄存器的内容如下：

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
COPYSEQPARAM			0	USTIME											

COPYSEQPARAM = Copy SEQPARAM bit value
(bits 15:13) in SequenceEvent Data register

USTIME = RAILn UnsequenceTime (in units of ms).

4.2.13 EnablePolarity

配置参数中的 **EnablePolarity** 字段用于指定是将每个电源使能或 GPO 配置为高电平有效还是低电平有效。这些寄存器的地址映射如下：

地址	容量	默认值	说明
0xE168	2	0x2004	电源轨 1 的极性使能
0xE16A	2	0x2008	电源轨 2 的极性使能
0xE16C	2	0x1804	电源轨 3 的极性使能
0xE16E	2	0x1802	电源轨 4 的极性使能
0xE170	2	0x1808	电源轨 5 的极性使能
0xE172	2	0x1810	电源轨 6 的极性使能
0xE174	2	0x1820	电源轨 7 的极性使能
0xE176	2	0x2010	电源轨 8 的极性使能
0xE178	2	0x2000	GPO1 的极性
0xE17A	2	0x2020	GPO2 的极性
0xE17C	2	0x2040	GPO3 的极性
0xE17E	2	0x2080	GPO4 的极性

此寄存器的内容如下：

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
POL	DEFAULT VALUES as specified previously														

POL	Meaning
0	Rail enable or GPO is active-low.
1	Rail enable or GPO is active-high.

4.2.14 SaveRailLog

配置参数中的 **SaveRailLog** 字段用于指定是否标记每个电源轨，以便在电源轨发生故障时将错误日志写入闪存。如果以这种方式标记电源轨，则该电源轨的关断将记录到非易失性存储器。在这种情况下，**NVERRLOG** 位 (**STATUS** 寄存器) 在器件初始化时设置为 1，并且必须在恢复正常序列发生器操作之前清零。有关详细说明，请参阅 [复位闪存错误日志](#) 一节。

此寄存器的内容如下：

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	RAIL8	RAIL7	RAIL6	RAIL5	RAIL4	RAIL3	RAIL2	RAIL1

RAILn	Meaning
0	Shutdown of this rail will not log this event.
1	Shutdown of this rail will log this event.

该寄存器的默认值为 0x0000。

4.2.15 ReferenceSelect

配置参数中的 **ReferenceSelect** 字段用于指定在 UCD9080 上使用的电压基准。所选基准可以是内部基准 (2.5V)，也可以是外部基准 (通过 V_{CC} (3.3V))。寄存器地址为 0xE186，内容如下：

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SELREF			0	0x8F2											

SELREF	Meaning
000	External reference selected (V_{CC})
001	Internal reference selected (2.5 V)

该寄存器的默认值为 0x08F2，它会选择外部基准。

4.2.16 LastUnusedSeq

配置参数中的 **LastUnusedSeq** 字段用于指定最后一个电源轨在不产生错误的情况下关闭的时间量。寄存器地址为 0xE18E，内容如下：

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LUTIME															

$$LUTIME = \text{Maximum value } USTIME + 255 \text{ (in units of ms)}$$

该寄存器的默认值为 0x08FF。

4.2.17 IgnoreGlitchAlarms

配置参数中的 **IgnoreGlitchAlarms** 字段可用于抑制记录干扰。有关完整说明，请参阅 [SLVS813](#) 中的“电压监测”部分。寄存器地址为 E06A，内容如下：

7	6	5	4	3	2	1	0
R8	R7	R6	R5	R4	R3	R2	R1

Rx = Rail #

Default = 0x00

Write 1 to bit location to suppress glitch alarms

4.2.18 IgnoreFlashErrorLog

配置参数中的 IgnoreFlashErrorLog 位可用于在闪存错误日志中存在条目时进行时序控制 (NVERRLOG 位设置为 1)。有关完整说明, 请参阅 [SLVS813](#) 中的“重置闪存错误日志”部分。寄存器地址为 E191, 内容如下:

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	IFEL

IFEL = Ignore flash error log

Default = 0x00

Write 1 to bit 0 location to ignore flash error log

4.2.19 Checksum

校验和字是配置参数中从 0xE000 到 0xE1FD 的字之和的 1 补码。校验和存储在器件参数部分的最后一个字中 (0xE1FF 为高字节, 0xE1FE 为低字节)。区域校验和 (和校验和) 以低字节/高字节 (“小端字节序”) 格式存储。

5 其他注意事项

5.1 嵌入式应用

由于 UCD9081 对典型的电源系统具有很大的控制能力，因此嵌入式应用在等待 I2C 命令时不会被暂停或延迟。嵌入式应用提供了各种机制，可在 I2C 通信中断时始终优先考虑本地时序控制和监测。这些机制主要是计时器。通常，如果计时器到期，I2C 主器件会收到 NACK，必须重新发出命令。

5.2 时序

UCD9081 的内部时序基准在整个过程、温度和电源电压范围内的变化幅度高达 $\pm 20\%$ 。以下各节中使用的延迟时间值列为标称值，用户必须应用适当的计时保护频带。

5.2.1 UCD9081 启动

加电后，无法立即与 UCD9081 建立 I2C 通信。加电时，UCD9081 对用户参数部分的内容执行多项操作。在此操作期间会产生复位延迟，数字 I/O 引脚处于高阻抗状态。此操作结束时，序列发生器应用程序启动，I2C 通信可能开始。

启动延迟可能因以下情况而异；

- 正常情况：UCD9081 对用户参数部分的内容执行校验和测试。如果测试通过，UCD9081 会将用户参数部分的内容与应用程序（或最后已知良好配置）参数部分的内容进行比较，以查看这些参数是否已更改。如果两个区域匹配，序列发生器应用程序会启动。如图 5-1 中所示，这通常需要 35ms 的时间

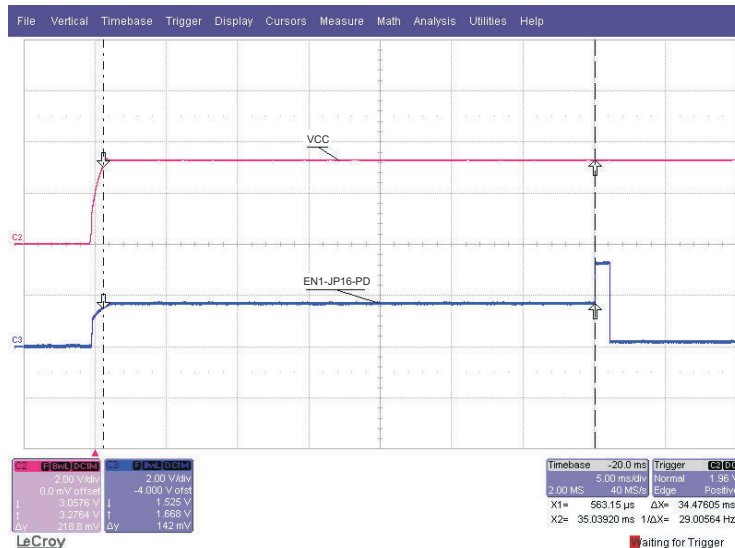


图 5-1. 上电延迟，无参数更改

- 配置更改通过校验和的情况：如果用户参数区域已更新，但器件尚未复位（请参阅 UCD9081 数据表 RESET 说明），则用户和应用程序参数区域不匹配。在下次复位后，会进行校验和测试。如果测试通过，则 UCD9081 会将用户参数复制到应用程序参数区域。此操作的标称时间为 102ms，如图 5-2 所示。后续上电（用户参数不变）属于正常类别。
- 配置更改未通过校验和的情况：如果校验和测试失败，则 UCD9081 会将应用程序参数复制回用户参数区域。此操作的标称时间为 102ms，如图 5-2 所示。

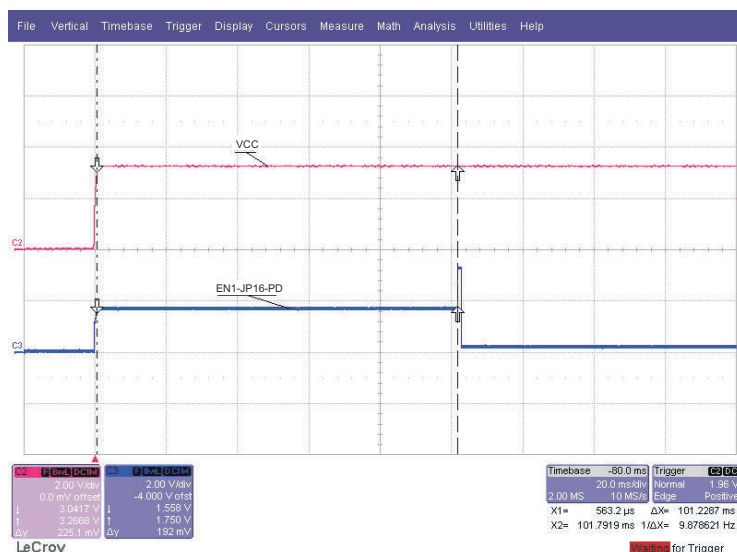


图 5-2. 上电延迟，有参数更改

5.2.2 擦除闪存后的时钟延展

在三个命令操作期间，UCD9081 在执行擦除时将 SCL 保持在低电平。在 WDATA 寄存器写入 0xBADC 后，UCD9081 执行擦除操作。有关向 WDATA 寄存器写入 0xBADC 的详细说明，请参阅以下 UCD9081 数据表部分。

- 重置闪存错误日志
- 配置 UCD9081
- 用户数据

擦除后，I2C 主器件可能会等待 SCL 被释放，也可能在发出另一个事务前等待大概 12ms。

5.2.3 位超时

针对以下列表中描述的位用例，UCD9081 会强制执行一个 $1/f_{SCL(MIN)}$ (100 μ s) 的最大位超时周期。必须满足这个位周期，否则 UCD9081 可能会退出事务，从而导致出现 NACK。对于以下列表中未描述的位用例，字节超时将如节 5.2.4 中所述应用。

- START 位 (图 5-3) : SCL 必须在 SDA 下降后的 100 μ s 内下降。

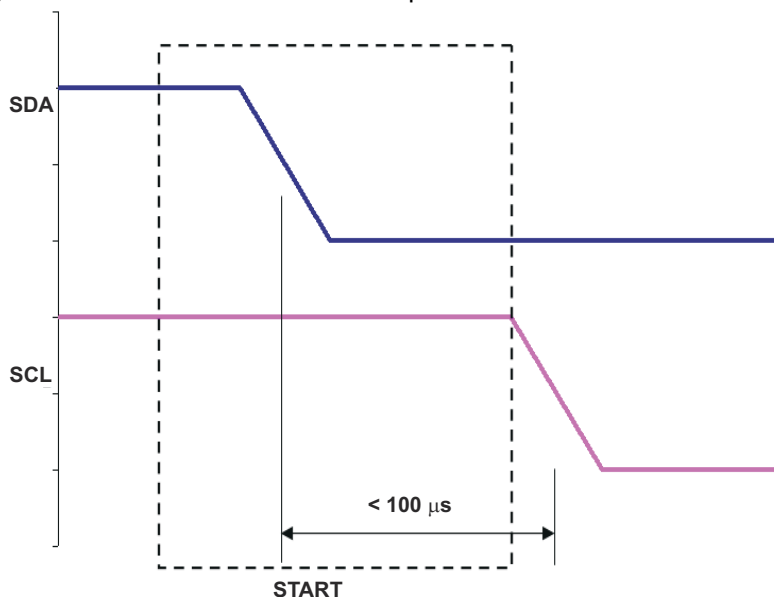


图 5-3. START 位要求

- 写入命令的第八个数据位 (图 5-4) : SCL 必须在 SCL 上升的 100 μ s 内下降。

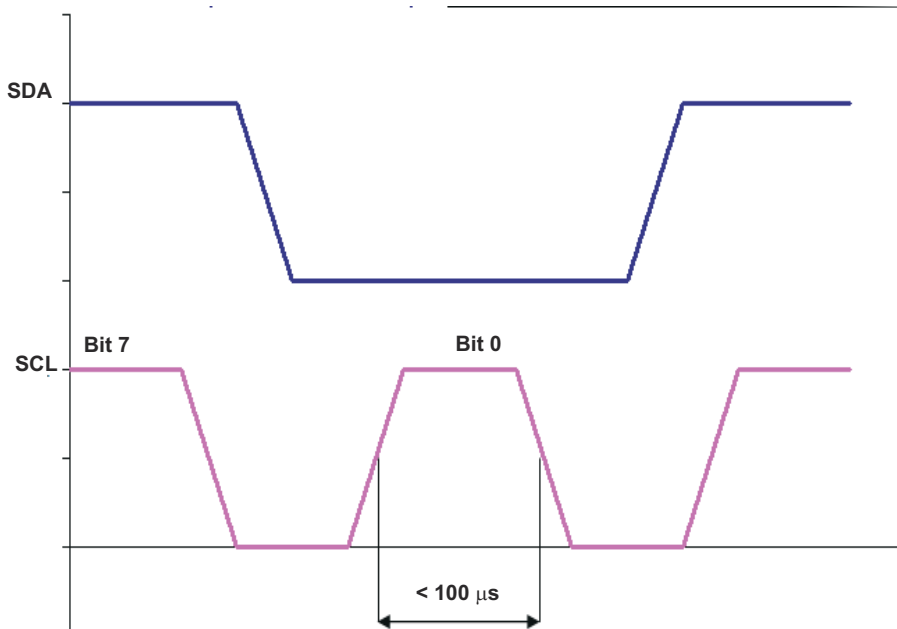


图 5-4. 写入命令的第八位要求

- 从器件地址命令的读取/写入位 (图 5-5) : SCL 必须在 SCL 上升的 100 μ s 内下降。

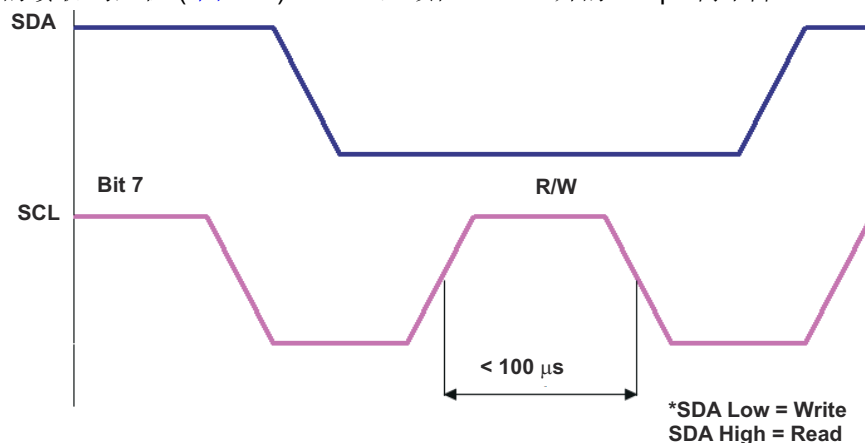


图 5-5. 读取/写入位要求

- 确认位 (图 5-6) : 在每个字节期间, 确认 (ACK) 或未确认 (NACK) 发生在 SCL 的第九个上升沿。在 ACK 或 NACK 之后, SCL 必须在 SCL 上升的 100 μ s 内下降。

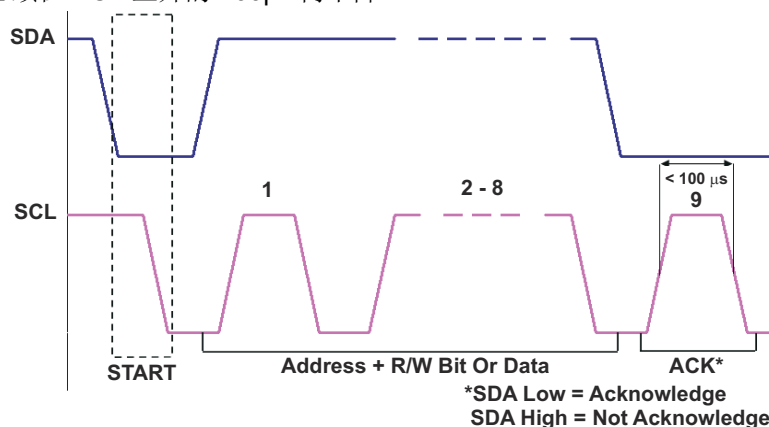


图 5-6. 确认位要求

5.2.4 字节或事务超时

UCD9081 在 I2C 主器件上强制执行大概 5ms 的最大字节或事务超时周期。该时间是从前一个字节下降 SDA (在 START 期间, 或重复 START 期间) 到当前字节下降 SDA (在 START 期间, 或重复 START 期间) 测得的。请参阅图 5-7。当 UCD9081 将 SCL 保持在低电平 (请参阅节 5.2.2) 以插入额外的等待状态 (即, 字节计时器被复位并且在延长期间不会过期) 时, 不会强制执行此超时。

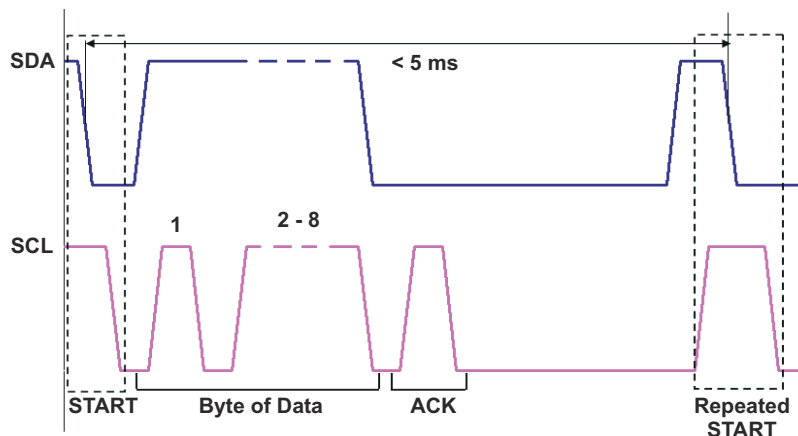


图 5-7. 字节超时要求

此外，当一个新位 (SCL 的上升沿) 到达时，字节超时周期被复位。因此，除了节 5.2.3 中描述四个异常外，字节超时周期也适用。

6 参考文献

1. [UCD9081 具有错误记录功能的 8 通道电源序列发生器和监测器 数据表](#)
2. [UCD9081 电源序列发生器和监测器 EVM 用户指南](#)
3. [使用 UCD9081 进行高级时序控制和监测 用户指南](#)

7 修订历史记录

注：以前版本的页码可能与当前版本的页码不同

Changes from Revision B (November 2010) to Revision C (May 2023)	Page
• 更新了整个文档中的表格、图和交叉参考的编号格式.....	1
• 在地址行 E040 和 E060 上添加了粗体.....	17
• 添加了节 4.2.1 GpDir	18
• 添加了节 4.2.2 NegateEnablePolarity	18
• 添加了节 4.2.3 SeqEventPending	18

重要声明和免责声明

TI“按原样”提供技术和可靠性数据（包括数据表）、设计资源（包括参考设计）、应用或其他设计建议、网络工具、安全信息和其他资源，不保证没有瑕疵且不做任何明示或暗示的担保，包括但不限于对适销性、某特定用途方面的适用性或不侵犯任何第三方知识产权的暗示担保。

这些资源可供使用 TI 产品进行设计的熟练开发人员使用。您将自行承担以下全部责任：(1) 针对您的应用选择合适的 TI 产品，(2) 设计、验证并测试您的应用，(3) 确保您的应用满足相应标准以及任何其他功能安全、信息安全、监管或其他要求。

这些资源如有变更，恕不另行通知。TI 授权您仅可将这些资源用于研发本资源所述的 TI 产品的应用。严禁对这些资源进行其他复制或展示。您无权使用任何其他 TI 知识产权或任何第三方知识产权。您应全额赔偿因在这些资源的使用中对 TI 及其代表造成的任何索赔、损害、成本、损失和债务，TI 对此概不负责。

TI 提供的产品受 [TI 的销售条款](#) 或 [ti.com](#) 上其他适用条款/TI 产品随附的其他适用条款的约束。TI 提供这些资源并不会扩展或以其他方式更改 TI 针对 TI 产品发布的适用的担保或担保免责声明。

TI 反对并拒绝您可能提出的任何其他或不同的条款。

邮寄地址：Texas Instruments, Post Office Box 655303, Dallas, Texas 75265

Copyright © 2023，德州仪器 (TI) 公司