

产品特性

16单元FIFO用于记录事件

10个可配置I/O支持如下功能

键盘解码支持最大5 x 5的矩阵

按键/释放中断

GPIO功能

GPI支持可选的中断级

100 kΩ或300 kΩ上拉电阻

300 kΩ下拉电阻

GPO支持推挽或开漏输出

可编程逻辑模块

脉冲发生器

周期和开启时间

30秒以上(125 ms增量)

最多255 ms(1 ms增量)

复位发生器

I²C接口支持增强快速模式(Fm+)，频率最高可达1 MHz

开漏中断输出

16引脚1.59 mm x 1.59 mm WLCSP封装

应用

键盘输入和输入/输出扩展功能

智能手机、遥控器和相机

医疗保健、工业和仪器仪表

概述

ADP5586是一款带10个I/O端口扩展器，内置键盘矩阵解码器、可编程逻辑、复位发生器和脉冲发生器。输入/输出扩展器IC适用于便携式设备(手机、遥控器和相机)及非便携式应用(医疗保健、工业和仪器仪表)，可用于增加处理器可用的I/O数量，或者通过接口连接器减少前面板设计所需的I/O数量。

ADP5586处理所有按键扫描和解码，并通过一条中断线通知主处理器有新的按键事件发生。GPI变化和逻辑变化也

功能框图

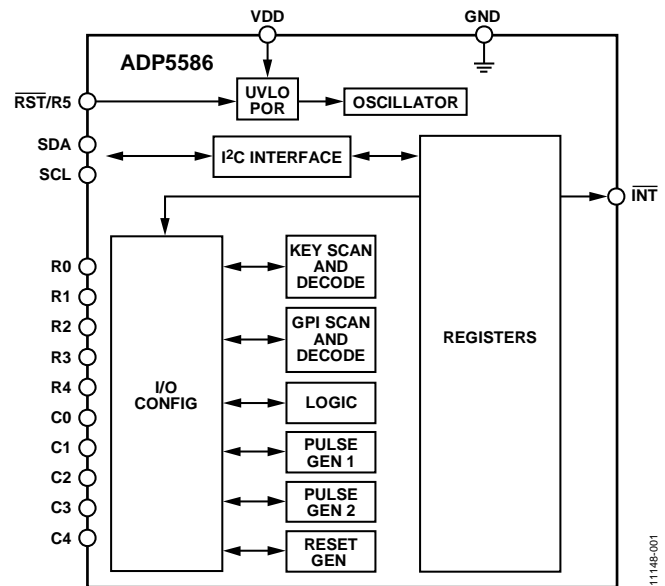


图1.

可以通过FIFO记录为事件，从而无需监控不同的寄存器来判断事件变化。ADP5586配有一个FIFO，它最多可以存储16个事件。处理器可以通过I²C兼容型接口回读事件。

ADP5586使主处理器不必监控键盘，从而降低功耗和/或提高处理器带宽，以便执行其它功能。

可编程逻辑功能支持将常用逻辑要求集成为GPIO扩展器的一部分，从而节省电路板面积和成本。

Rev. 0

[Document Feedback](#)

Information furnished by Analog Devices is believed to be accurate and reliable. However, no responsibility is assumed by Analog Devices for its use, nor for any infringements of patents or other rights of third parties that may result from its use. Specifications subject to change without notice. No license is granted by implication or otherwise under any patent or patent rights of Analog Devices. Trademarks and registered trademarks are the property of their respective owners.

One Technology Way, P.O. Box 9106, Norwood, MA 02062-9106, U.S.A.
Tel: 781.329.4700 ©2013 Analog Devices, Inc. All rights reserved.
[Technical Support](#) www.analog.com

ADI中文版数据手册是英文版数据手册的译文，敬请谅解翻译中可能存在的语言组织或翻译错误，ADI不对翻译中存在的差异或由此产生的错误负责。如需确认任何词语的准确性，请参考ADI提供的最新英文版数据手册。

目录

产品特性	1	事件FIFO	9
应用	1	键扫描控制	10
功能框图	1	GPI输入	13
概述	1	GPO输出	13
修订历史	2	逻辑模块	14
技术规格	3	复位模块	15
I ² C时序规格	4	中断	15
绝对最大额定值	5	脉冲发生器	16
热阻	5	寄存器接口	17
ESD警告	5	寄存器映射	19
引脚配置和功能描述	6	寄存器描述	21
工作原理	7	应用原理图	41
器件使能	8	外形尺寸	42
器件概述	8	订购指南	42
功能描述	9		

修订历史

2013年3月—修订版0：初始版

技术规格

除非另有说明, VDD = 1.8 V至3.3 V, $T_A = T_J = -40^{\circ}\text{C}$ 至 $+85^{\circ}\text{C}$ 。¹

表1.

参数	符号	测试条件/注释	最小值	典型值	最大值	单位
电源电压						
VDD输入电压范围	VDD		1.65		3.6	V
欠压闭锁阈值	UVLO _{VDD}	UVLO有效, VDD下降 UVLO无效, VDD下降	1.2	1.3	1.6	V
电源电流						
待机电流	I _{STNBY}	VDD = 1.65 V VDD = 3.3 V		1	4	μA
工作电流(按下一个键)	I _{SCAN1} I _{SCAN2}	扫描 = 10 ms, CORE_FREQ = 50 kHz, 扫描有效, 300 kΩ上拉电阻, VDD = 1.65 V 扫描 = 10 ms, CORE_FREQ = 50 kHz, 扫描有效, 300 kΩ上拉电阻, VDD = 3.3 V		30	40	μA
上拉、下拉电阻						
上拉						
选项1			50	100	150	kΩ
选项2			150	300	450	kΩ
下拉			150	300	450	kΩ
输入逻辑电平(RST、SCL、SDA、R0、R1、R2、R3、R4、R5、C0、C1、C2、C3、C4)						
输入电压						
逻辑低电平	V _{IL}				0.3 × VDD	V
逻辑高电平	V _{IH}		0.7 × VDD			V
输入漏电流(每引脚)	V _{I-LEAK}			0.1	1	μA
推挽输出逻辑电平(R0、R1、R2、R3、R4、R5、C0、C1、C2、C3、C4)						
输出电压						
逻辑低电平	V _{OL1} V _{OL2}	吸电流 = 10 mA, 最多5个GPIO 同时有效 吸电流 = 10 mA, 所有GPIO同时有效			0.4 0.5	V
逻辑高电平	V _{OH}	源电流 = 5 mA	0.7 × VDD			V
逻辑高电平输出漏电流(每引脚)	V _{OH-LEAK}			0.1	1	μA
开漏输出逻辑电平(INT、SDA)						
输出电压						
逻辑低电平						
INT	V _{OL3}	I _{SINK} = 10 mA			0.4	V
SDA	V _{OL4}	I _{SINK} = 20 mA			0.4	V
逻辑高电平输出漏电流(每引脚)	V _{OH-LEAK}			0.1	1	μA
逻辑传播延迟				125	300	ns
触发器(FF)保持时间 ²				0		ns
FF建立时间 ²				175		ns
GPIO去抖 ²					70	μs
内部振荡器频率 ³	OSC _{FREQ}		720	800	880	kHz

¹ 所有极端温度下的限值采用标准统计质量控制(SQC)通过相关性予以保证。典型值的条件: $T_A = 25^{\circ}\text{C}$, VDD = 1.8 V。

² 通过设计保证。

³ 所有定时器均参考基本振荡器, 并具有相同的±10%精度。

ADP5586

I²C时序规格

表2.

参数	说明	最小值	最大值	单位
I ² C时序规格				
UVLO/RST无效到I ² C访问的延迟时间			60	μs
f _{SCL}	SCL时钟频率	0	1000	kHz
t _{HIGH}	SCL高电平时间	0.26		μs
t _{LOW}	SCL低电平时间	0.5		μs
t _{SU; DAT}	数据建立时间	50		ns
t _{HD; DAT}	数据保持时间	0		μs
t _{SU; STA}	重复起始建立时间	0.26		μs
t _{HD; STA}	起始/重复起始保持时间	0.26		μs
t _{BUF}	停止和起始条件的总线空闲时间	0.5		μs
t _{SU; STO}	停止条件的建立时间	0.26		μs
t _{VD; DAT}	数据有效时间		0.45	μs
t _{VD; ACK}	数据有效应答		0.45	μs
t _R	SCL和SDA的上升时间		120	ns
t _F	SCL和SDA的下降时间		120	ns
t _{SP}	抑制尖峰的脉冲宽度	0	50	ns
C _B ¹	各条总线的容性负载		550	pF

¹ C_B是一条总线的总电容(单位: pF)。

时序图

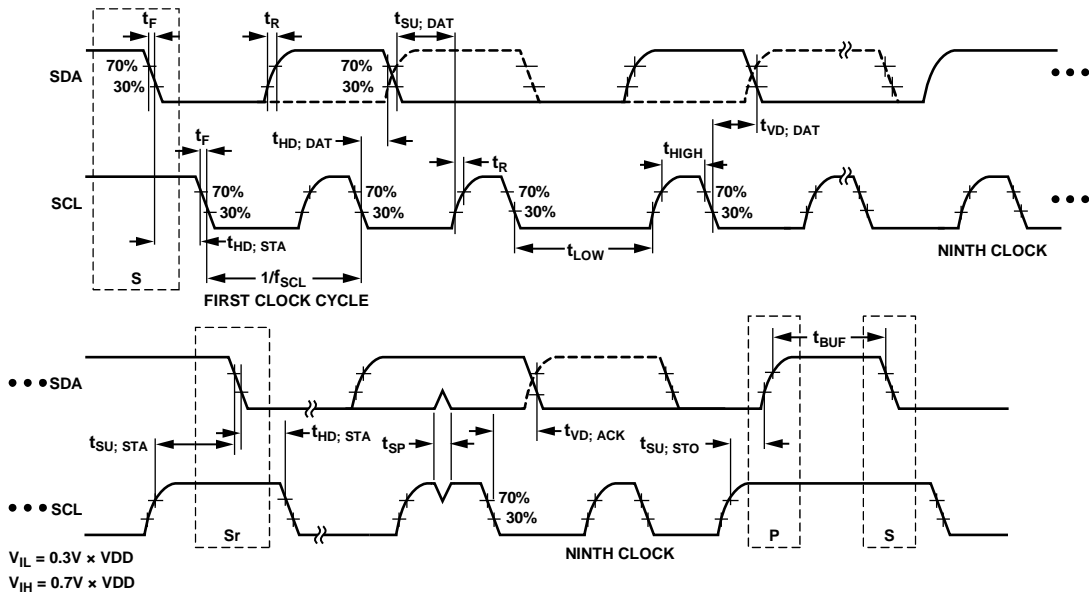


图2. I²C接口时序图

11148-002

绝对最大额定值

表3.

参数	额定值
VDD至GND	-0.3 V至+4 V
SCL, SDA, \overline{RST} , \overline{INT} , R0, R1, R2, R3, R4, C0, C1, C2, C3, C4	-0.3 V至(VDD + 0.3 V)
温度范围	
工作(环境)	-40°C至+85°C ¹
工作(结温)	-40°C至+125°C
存储	-65°C至+150°C

¹ 在功耗高、热阻差的应用中，可能需要降低最大环境温度。最大环境温度($T_{A(MAX)}$)取决于最大工作结温($T_{J(MAXOP)} = 125^{\circ}\text{C}$)、器件的最大功耗($P_{D(MAX)}$)和应用中器件/封装的结至环境热阻(θ_{JA})，表示为下式： $T_{A(MAX)} = T_{J(MAXOP)} - (\theta_{JA} \times P_{D(MAX)})$ 。

注意，超出上述绝对最大额定值可能会导致器件永久性损坏。这只是额定最值，并不能以这些条件或者在任何其它超出本技术规范操作章节中所示规格的条件下，推断器件能否正常工作。长期在绝对最大额定值条件下工作会影响器件的可靠性。

绝对最大额定值仅适合单独应用，但不适合组合使用。除非另有规定，所有其它电压均以GND为参考。

热阻

θ_{JA} 针对最差条件，即器件焊接在印刷电路板(PCB)上以实现表贴封装。

表4.

热阻	θ_{JA}	单位
16引脚WLCSP	62	$^{\circ}\text{C}/\text{W}$
最大功耗	70	mW

ESD警告



ESD(静电放电)敏感器件。

带电器件和电路板可能会在没有察觉的情况下放电。尽管本产品具有专利或专有保护电路，但在遇到高能量ESD时，器件可能会损坏。因此，应当采取适当的ESD防范措施，以避免器件性能下降或功能丧失。

引脚配置和功能描述

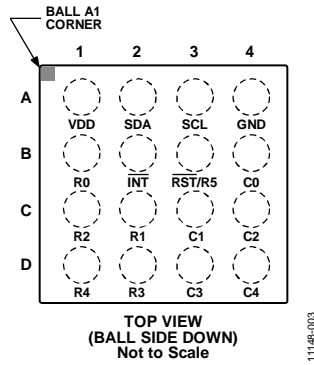
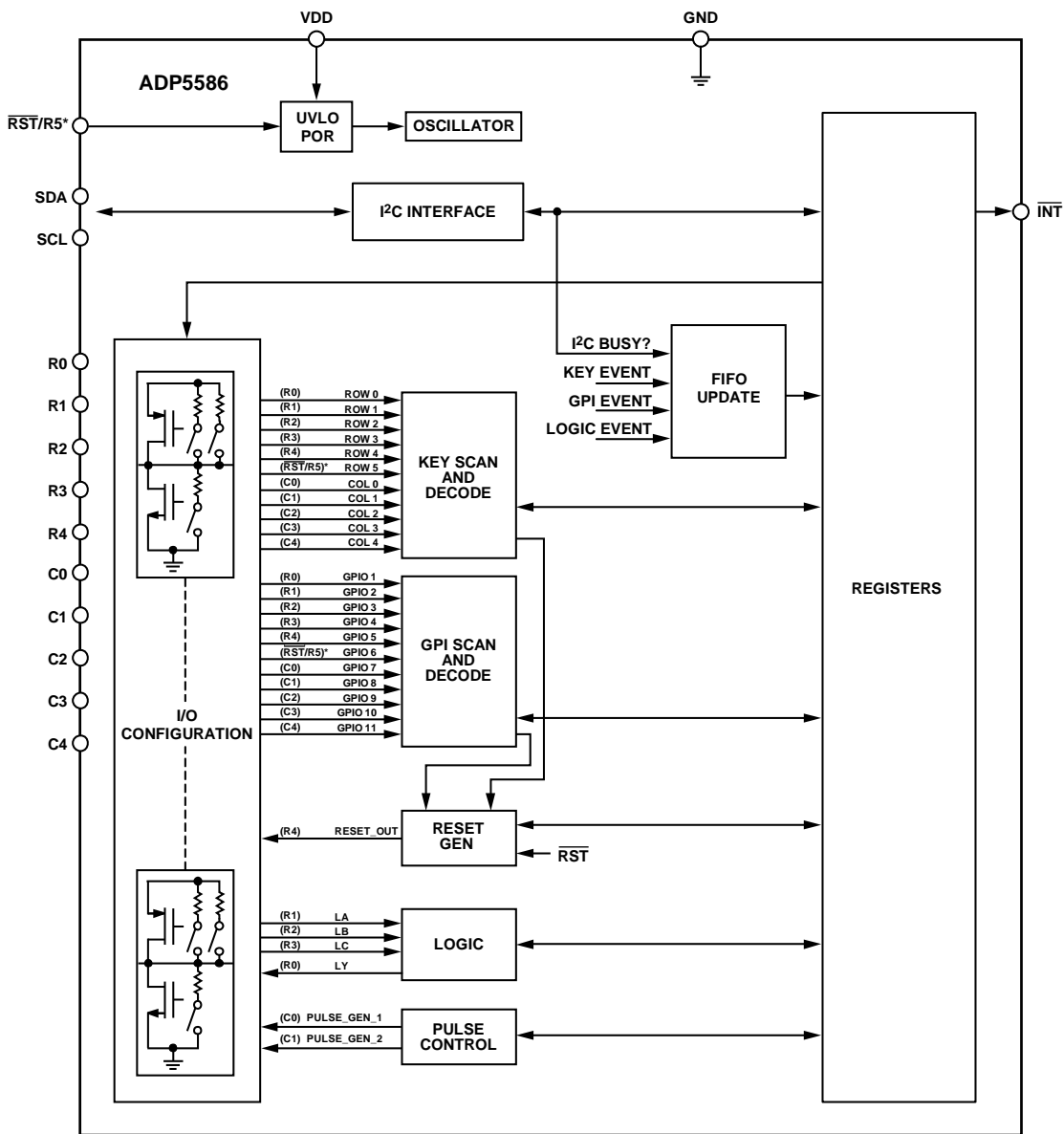


图3. 引脚配置

表5. 引脚功能描述

引脚编号	引脚名称	说明
A1	VDD	电源电压输入。
A2	SDA	I ² C数据输入/输出。
A3	SCL	I ² C时钟输入。
A4	GND	地。
B1	R0	GPIO 1(GPIO备选功能: 逻辑模块输出LY)。此引脚在键盘模式下用作行0。
B2	$\overline{\text{INT}}$	开漏中断输出。
B3	$\overline{\text{RST/R5}}$	输入复位信号($\overline{\text{RST}}$)。复位信号功能适用于除ADP5586ACBZ-01-R7以外的所有型号。 GPIO 6/行5 (R5)。此功能仅适用于ADP5586ACBZ-01-R7型号。
B4	C0	GPIO 7(GPIO备选功能: PULSE_GEN_1)。此引脚在键盘模式下用作列0。
C1	R2	GPIO 3(GPIO备选功能: 逻辑模块输入LB)。此引脚在键盘模式下用作行2。
C2	R1	GPIO 2(GPIO备选功能: 逻辑模块输入LA)。此引脚在键盘模式下用作行1。
C3	C1	GPIO 8(GPIO备选功能: PULSE_GEN_2)。此引脚在键盘模式下用作列1。
C4	C2	GPIO 9。此引脚在键盘模式下用作列2。
D1	R4	GPIO 5(GPIO备选功能: RESET_OUT)。此引脚在键盘模式下用作行4。
D2	R3	GPIO 4(GPIO备选功能: 逻辑模块输入LC)。此引脚在键盘模式下用作行3。
D3	C3	GPIO 10。此引脚在键盘模式下用作列3。
D4	C4	GPIO 11。此引脚在键盘模式下用作列4。

工作原理



*R5 AVAILABLE ON ADP5586ACBZ-01-R7 ONLY.

图4. 内部框图

11148-004

ADP5586

器件使能

对VDD施加足够大的电压并用逻辑高电平驱动 $\overline{\text{RST}}$ 引脚时，ADP5586启动进入待机模式，所有设置均为默认值。用户可通过I²C接口配置器件。当 $\overline{\text{RST}}$ 引脚为低电平时，ADP5586进入复位状态，所有设置恢复默认值。 $\overline{\text{RST}}$ 引脚有一个去抖滤波器。

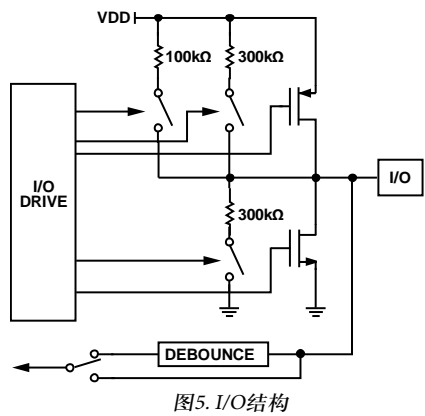
使用ADP5586ACBZ-01-R7型号时， $\overline{\text{RST}}$ 引脚用作额外的行引脚(R5)。要复位器件而不使用复位引脚，可将VDD拉低到UVLO阈值以下，或将SW_RESET位设为1(寄存器0x3D的位2)。

器件概述

ADP5586包含10个多重配置输入/输出引脚。每个引脚都可使能器件以执行不同的功能，如下所示：

- 键盘矩阵解码(最多5列乘5行矩阵)
- 通用I/O扩展(最多10路输入/输出)
- 复位发生器
- 逻辑功能构建模块(最多3路输入和1路输出)
- 2个脉冲发生器

所有10个输入/输出引脚都有图5所示的I/O结构。



每个I/O都可以通过100 kΩ或300 kΩ电阻上拉，或通过300 kΩ电阻下拉。对于逻辑输出驱动，推挽型输出的每个I/O具有5 mA PMOS源电流和10 mA NMOS吸电流，开漏输出则不支持5 mA PMOS源电流。对于逻辑输入应用，每个I/O可以直接采样，或通过去抖滤波器采样。

图5所示的I/O结构支持所有GPI和GPO功能，以及PWM和时钟分频功能。对于键盘矩阵扫描和解码，扫描电路使用100 kΩ或300 kΩ电阻上拉键盘行引脚，10 mA NMOS吸电流用于将键盘列引脚接地(关于键盘解码的详细信息参见“键扫描控制”部分)。

通过I²C接口对一系列内部寄存器进行编程，可以配置器件。器件状态的反馈和待处理的中断可利用 $\overline{\text{INT}}$ 引脚反映给外部处理器。

ADP5586提供三种特性集。表6列出了ADP5586各型号提供的选项。关于供货情况和/或其它配置，请联系您当地ADI公司的现场应用工程师。

表6. 器件各型号的矩阵选项¹

型号	说明
ADP5586ACBZ-00-R7	启动时GPIO下拉 5行 x 5列矩阵
ADP5586ACBZ-01-R7	GPIO增加行5 6行 x 5列矩阵
ADP5586ACBZ-03-R7	备选I ² C地址(0x30) 5行 x 5列矩阵

¹ 关于供货情况或本表未显示的配置，请联系ADI公司。

功能描述

事件FIFO

详细讨论ADP5586的各个模块之前，必须了解其特色配置——事件FIFO的功能。事件FIFO(寄存器0x03至寄存器0x12)最多可以记录16个事件。默认情况下，FIFO主要记录键事件，如键按下和键释放等。然而，也可以配置通用输入(GPI)和逻辑活动在FIFO上产生事件信息。事件计数EC[4:0](寄存器0x02的位[4:0])由5位组成，与FIFO配合工作，以使用户知道FIFO中存储了多少事件。

FIFO由16个8位单元组成。各单元的位[6:0]存储事件标识符，位7存储事件状态。用户可以从FIFO_1至FIFO_16的任意寄存器读取FIFO的顶部单元。ADP5586具有FIFO寄存器的多个副本，允许用一个I²C突发读取操作读取完整FIFO。

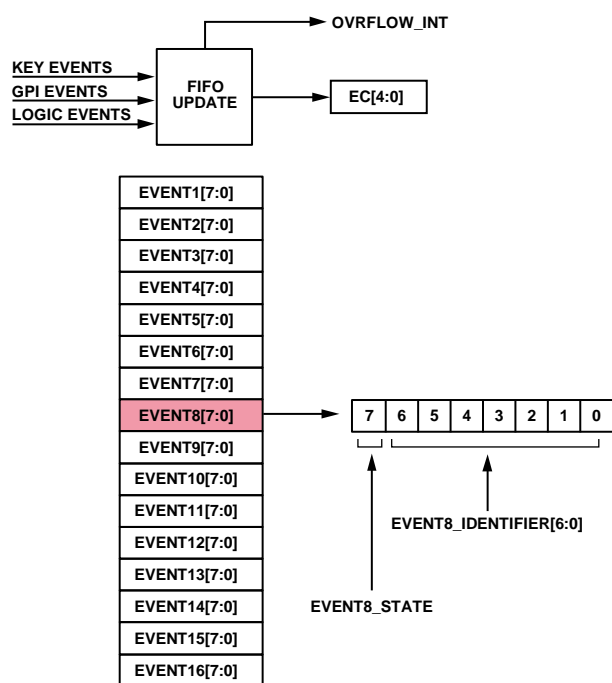


图6. Eventx[7:0]位细分

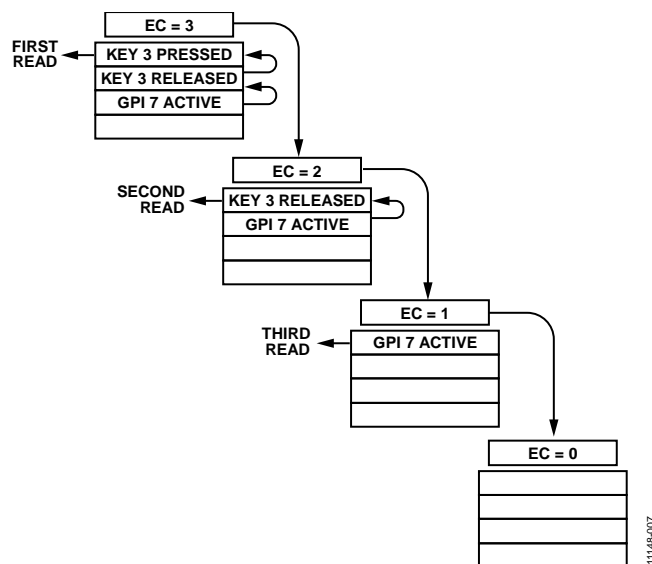


图7. FIFO操作

FIFO寄存器始终指向FIFO的顶部(即EVENT1[7:0]的位置)。如果用户试图回读FIFO中的任何位置，数据始终是从FIFO顶部获得。这样可确保事件只能按发生的先后顺序回读，从而保证FIFO系统的完整性。

如上所述，ADP5586的一些片上功能可以设置为在FIFO上产生事件。FIFO更新控制模块管理FIFO更新。若有I²C处理正在访问任何FIFO地址位置，更新将暂停，直到I²C处理完成为止。

在外部处理器读取并清除FIFO之前，如果产生了16个以上的事件，则发生FIFO溢出事件。

发生溢出事件时，溢出中断状态位置1(OVRFLOW_INT，寄存器0x01的位2)。如果使能了溢出中断，则将产生中断，告知处理器发生了16个以上的事件。

键扫描控制

一般特性

利用PIN_CONFIG_A、PIN_CONFIG_B和PIN_CONFIG_C寄存器(寄存器0x3A至0x3C), 10个输入/输出引脚可配置为解码最多包含25个开关(5×5矩阵)的键盘。也可以配置更小的矩阵, 不用的行和列引脚用于其它I/O功能。

R0至R4 I/O引脚构成键盘矩阵的行。C0至C4 I/O引脚构成键盘矩阵的列。用作行的引脚由内部300 kΩ(或100 kΩ)电阻上拉。用作列的引脚由内部NMOS吸电流驱动到低电平。

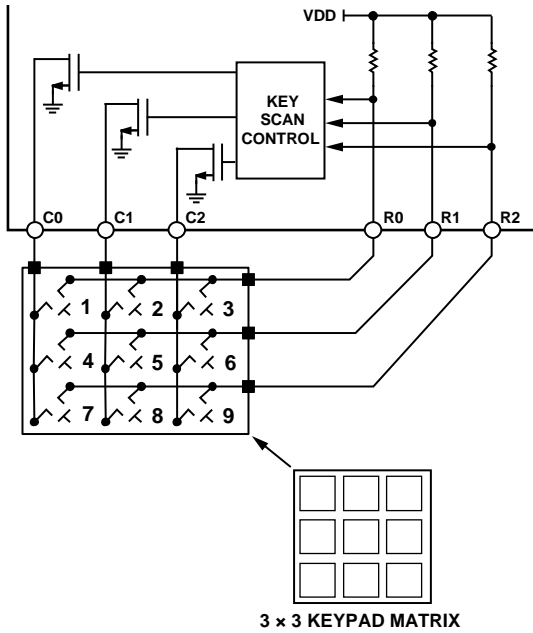


图8. 简化键扫描模块

图8是键扫描模块的简化示意图, 它使用3个行引脚和3个列引脚, 连接成一个小九开关(3×3)键盘矩阵。键扫描器空闲时, 行引脚被拉高, 列引脚被驱低。键扫描器检查行引脚, 查看它们是否为低电平。

按下矩阵中的开关6时, R1连接到C2。键扫描电路检测到一个行引脚被拉低时, 键扫描周期即开始。键扫描会将所有列引脚驱动到高电平, 然后将各列引脚驱动到低电平, 一次一个, 并检测是否有某个行引脚为低电平。所有行/列对都会被扫描, 因此按下多个键时也能检测到。

为防止毛刺或非常短的按键被记录为有效按键, 键扫描器要求按键持续2个扫描周期。各扫描周期之间有一个等待时间, 因此, 键必须按下并至少保持此等待时间, 才能被记录为有效按键。如果键被持续按下, 键扫描器将继续扫描, 并等待直到键被释放。

释放开关6时, R1与C2之间的连接断开, R1被拉高。键扫描器要求键释放时间为2个扫描周期, 因为键的释放并不一定与键扫描器同步。为使一个键被记录为释放, 最多可能需要2个完整的等待/扫描周期。键被记录为释放, 并且没有其它键被按下时, 键扫描器回到空闲模式。

在本数据手册的其余部分, 键的按下/释放将被简单地表示为图中的一个逻辑信号。逻辑高电平表示键状态为按下, 逻辑低电平表示释放。这样, 描述键事件时就无需绘制各行/列信号。

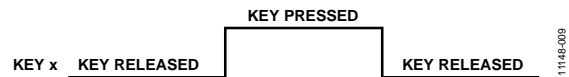


图9. 逻辑低电平: 键释放; 逻辑高电平: 键按下

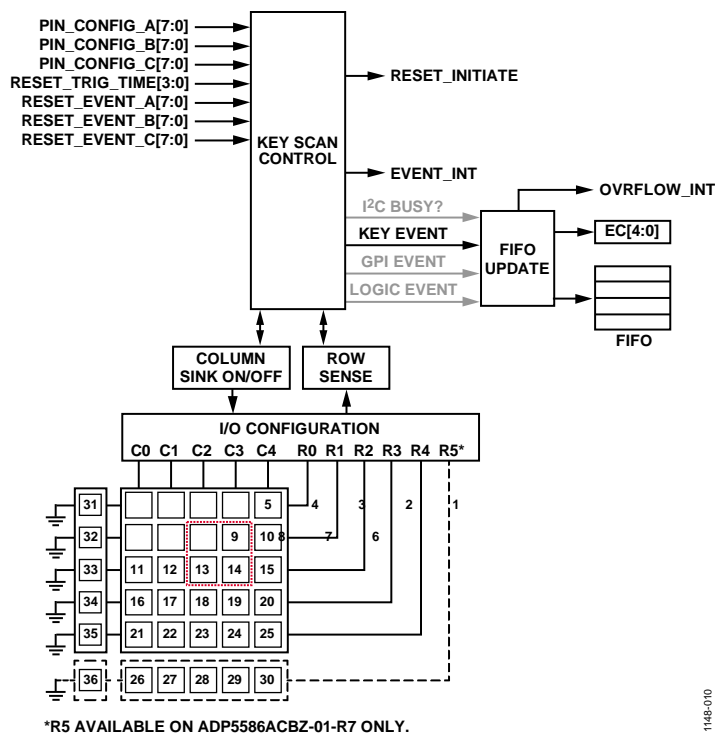


图10. 键扫描模块详图

图10显示了键扫描模块及其相关控制和状态信号的详细示意图。所有行和列引脚都使用时，可以扫描由25个唯一键组成的矩阵。

使用PIN_CONFIG_A[5:0]和PIN_CONFIG_B[4:0]寄存器(分别为寄存器0x3A和寄存器0x3B)配置键盘解码的I/O。每个键开关上的编号标签表示该开关按下时记录的事件标识符。如果配置所有行/列引脚，则可以在FIFO上观察到所有25个键标识符。

如果配置较小的 2×2 矩阵，例如使用C2、C3列引脚和R1、R2行引脚，则在FIFO上只能观察到4个事件标识符(8、9、13和14)，如图10所示。

默认情况下，ADP5586在FIFO上记录键按下和释放事件。图11说明了按下和释放单个键时发生的事情。最开始，键扫描器处于空闲状态。按下键3时，扫描器开始扫描所有已配置的行/列对。经过扫描等待时间后，扫描器再次扫描所有已配置的行/列对，检测到键3仍被按下，因而设置EVENT_INT中断位(寄存器0x01的位0)。然后，事件计数器EC[4:0](寄存器0x02的位[4:0])递增至1；FIFO的EVENT1_IDENTIFIER[6:0]更新，其事件标识符设置为3，并且EVENT1_STATE位置1，表示键按下事件。

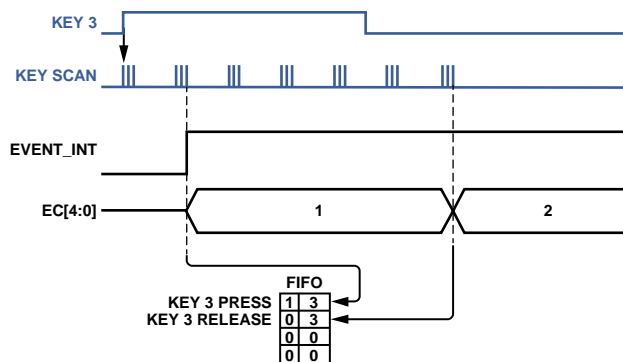


图11. 按下和释放事件

在键保持按下状态的期间，键扫描器继续扫描/等待周期。如果扫描器检测到该键已被释放并持续2个连续扫描周期，则事件计数器EC[4:0]递增至2；FIFO的EVENT2_IDENTIFIER[6:0]更新，其事件标识符设置为3，并且EVENT2_STATE位置1，表示键释放事件。没有其它键被按下时，键扫描器回到空闲模式。

按下和释放键事件均可触发EVENT_INT中断(寄存器0x01的位0)。如图12所示,如果按下键3,则EVENT_INT置位,EC[4:0]更新,并且FIFO更新。在该键保持按下状态的期间,可以读取FIFO,事件计数器递减至0,EVENT_INT被清除。当该键最终被释放时,EVENT_INT置位,事件计数器递增,FIFO更新为释放事件信息。

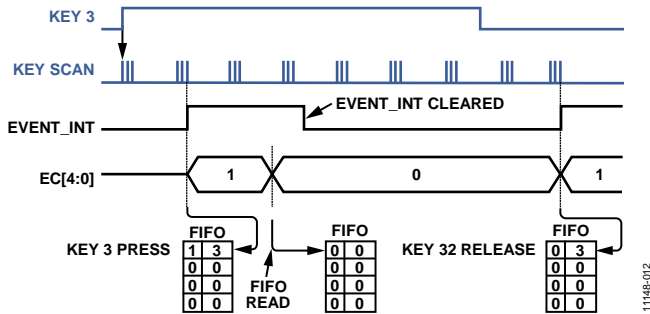


图12. 置位EVENT_INT中断键盘扩展

如图10所示,如果各行通过开关直接接地,就可以扩展键盘。按下R0与地之间的开关时,整行接地。当键扫描器完成扫描时,它一般会检测到键1至键5被按下,但ADP5586对这一独特的状况进行解码后,为其指定键事件31。还可以再指定5个键事件,使键盘大小扩展到30。然而,按下扩展键之一时,该行上的任何键都无法被检测。地键的激活导致共享该行的所有其它键都无法被检测。

预充电时间

在扫描序列中,一行按顺序扫描各列。各行/列组合均以KEY_POLL_TIME位(寄存器0x39的位[1:0])定义的速率进行测试。在各扫描时间内,各列的扫描时间由PRECHARGE_TIME位(寄存器0x39的位3)定义。如图13所示,由串联电阻(例如来自上拉/下拉电阻)和各列上的并联电容定义的电阻电容(RC)时间常数会影响按键事件的采样。

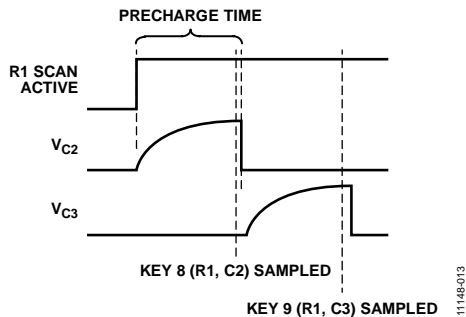


图13. 预充电时间

ADP5586在预充电时间快要结束时对行/列对的状态进行采样。通过延长此时间,就可以支持更大的RC时间常数。对于使用物理按钮的应用,RC时间常数一般不是问题,但若将外部继电器开关或多个外部复用器连接到各列,RC常数可能会增大。在行上使用较小的上拉电阻(寄存器0x3C的位7)可减小RC时间常数。

重影

重影是指这样一种现象:按下键盘矩阵上的某些键组合时,会检测到一个其它键的虚假有效读数。同时按下多行或多列上的三个或更多键时,会产生重影(参见图14)。在键盘矩阵上形成直角的键组合可能会引起重影。

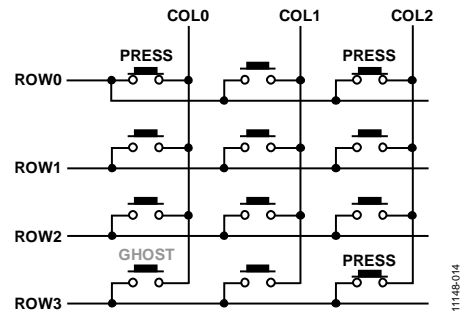


图14. 重影示例:在按键期间,由于行0、列0、列2和行3短路,列0/行3成为重影键

解决重影的办法是在键盘矩阵布局中考虑最有可能被同时按下的三键组合。按下一行或一列上的多个键不会引起重影。交错安排键,使它们不共享一列,也能避免重影。最常用的做法是将有可能被同时按下的键放在同一行或同一列。下面是一些可能被同时按下的键示例:

- 导航键和选择键的组合
- 导航键和空格键的组合
- 复位符合键,如CTRL + ALT + DEL

GPI输入

使用GPIO_INP_EN_A和GPIO_INP_EN_B寄存器(寄存器0x29和寄存器0x2A),可以将10条输入/输出线中的各条线配置为通用逻辑输入线。GPIO线可以配置为同时支持输入和输出。图15显示了GPI扫描检测模块及其相关控制和状态信号的详细示意图。

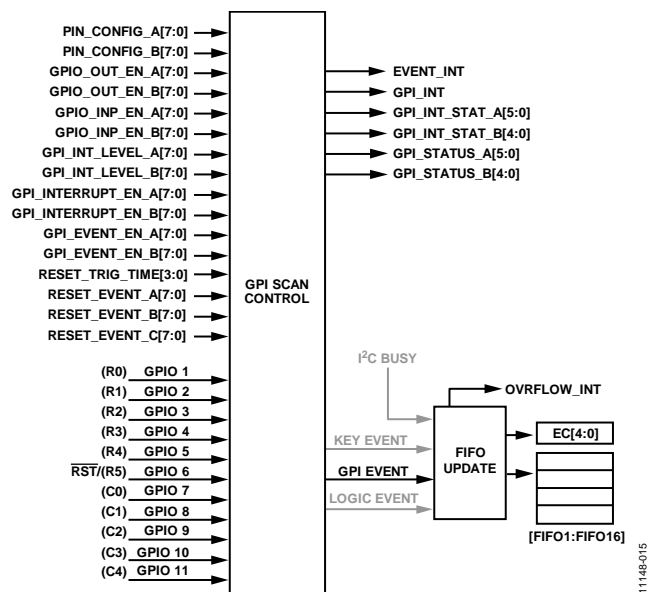


图15. GPI扫描检测模块

各GPI的当前输入状态可通过GPI_STATUS_x寄存器(寄存器0x15和寄存器0x16)回读。各GPI均可通过GPI_INTERRUPT_EN_x寄存器(寄存器0x1F和寄存器0x20)配置产生中断。中断状态存储在GPI_INT_STAT_x寄存器(寄存器0x13和寄存器0x14)。通过配置GPI_INT_LEVEL_x寄存器(寄存器0x1B和寄存器0x1C),可使GPI中断在正边沿或负边沿触发。触发任何GPI中断时,主GPI_INT中断位(寄存器0x01的位1)也会被触发。图16显示了单个GPI及其如何影响对应的状态和中断状态位。

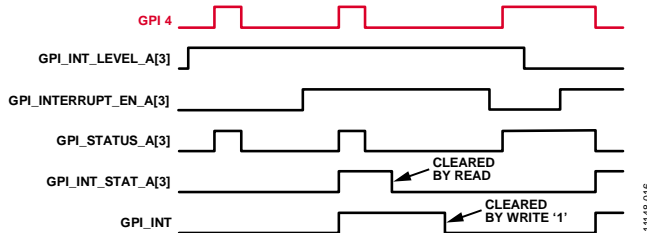


图16. 单个GPI示例

通过GPI_EVENT_EN_x寄存器(寄存器0x1D和寄存器0x1E),可配置GPI产生FIFO事件。这种模式下,GPI不产生GPI_INT中断。而是产生EVENT_INT中断(寄存器0x01的位0)。图17显示了多条GPI线及其对FIFO和事件计数EC[4:0]的影响。

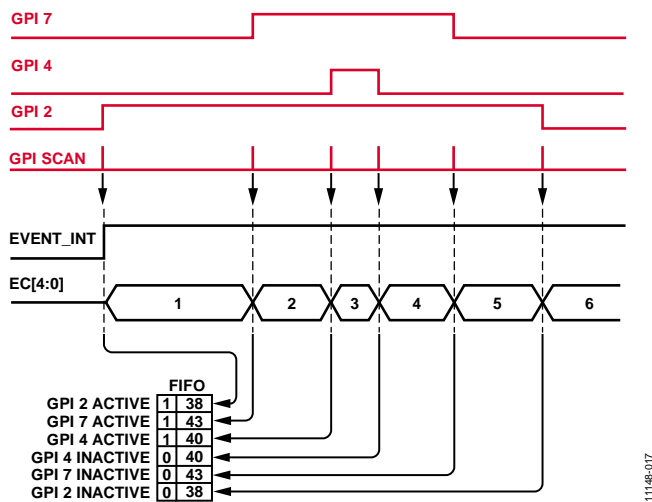


图17. 多个GPI示例

GPI扫描器保持空闲,直至检测到电平转换。然后,它扫描GPI输入并进行相应的更新。更新后,它立刻回到空闲状态,而不是像键扫描器那样扫描/等待。因此,在窄脉冲通过70 μs输入去抖滤波器后,GPI扫描器可以检测到窄脉冲的两个边沿。

GPO输出

使用GPIO_OUT_EN_A和GPIO_OUT_EN_B寄存器(寄存器0x27和寄存器0x28),可以将10条输入/输出线中的各条线配置为通用输出(GPO)线。GPIO线可以配置为同时支持输入和输出(I/O结构详图参见图5)。GPO配置和使用是在GPO_DATA_OUT_x和GPO_OUT_MODE_x寄存器(寄存器0x23至寄存器0x26)中设置。更多信息参见“寄存器描述”部分。

ADP5586

逻辑模块

ADP5586的多条输入/输出线可以用作输入和输出来实现某些通用逻辑功能。

R1、R2和R3输入/输出引脚可用作输入，R0输入/输出可用作逻辑模块的输出。使用R1、R2和R3输入线时，必须使能GPIO_4_INP_EN、GPIO_3_INP_EN和GPIO_2_INP_EN位(寄存器0x29的位[3:1])以接受输入。

R0引脚用作逻辑模块的输出时，必须使能GPIO_1_OUT_EN位(寄存器0x27的位0)。

逻辑模块的输出可以配置产生中断，也可以配置产生FIFO事件。

图19显示了逻辑模块内部组成的详图，说明了可以实现的逻辑功能。

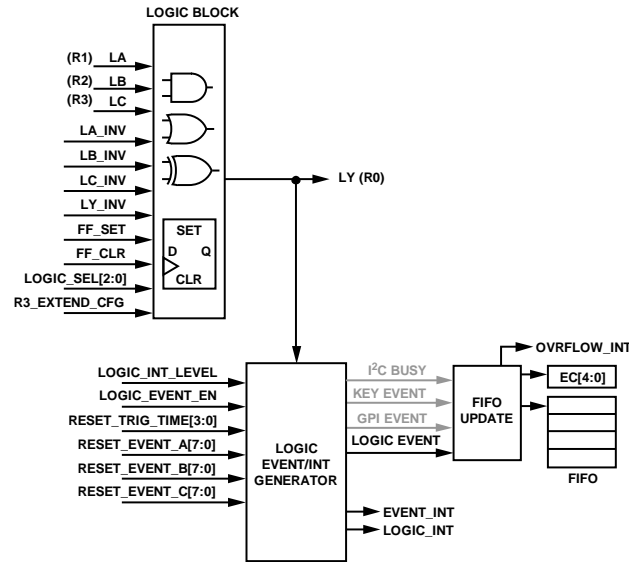


图18. 逻辑模块概览

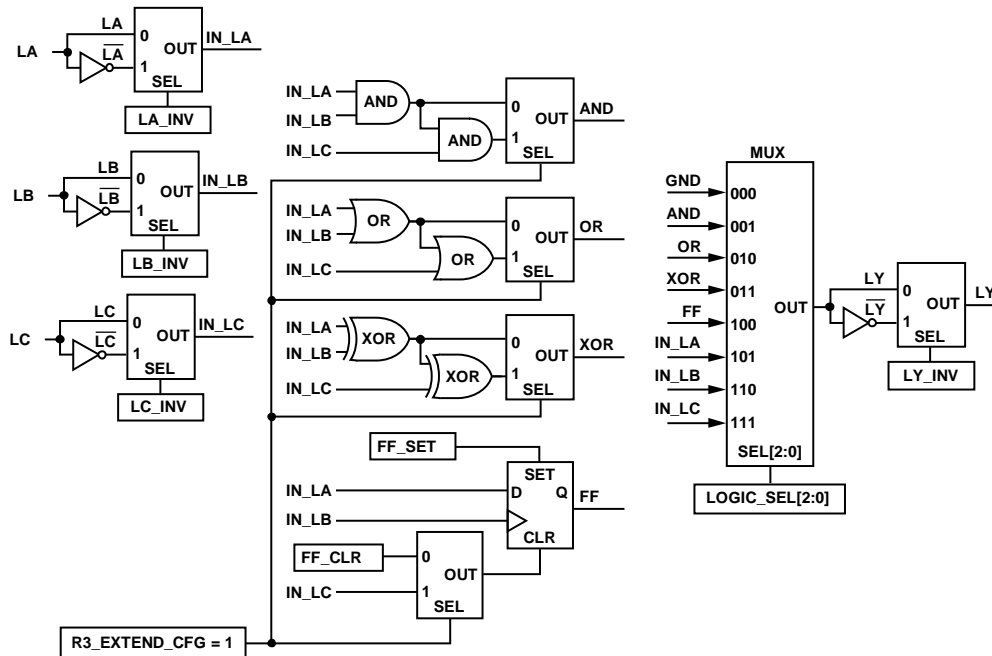


图19. 逻辑模块内部组成

复位模块

ADP5586有一个复位模块，如果同时检测到某些事件，它可以产生复位条件。针对RESET_OUT最多可以设置三个复位触发事件。事件扫描控制模块监控这些事件在RESET_TRIG_TIME[3:0](寄存器0x2E的位[5:2])设置的期间内是否存在。如果存在，就会将复位启动信号发送至复位发生器模块。所产生的复位信号脉冲宽度是可编程。

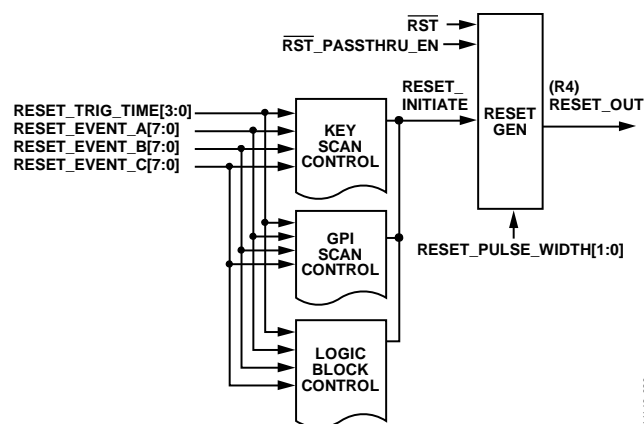


图20. 复位模块

RESET_OUT信号使用R4 I/O引脚作为输出，必须通过GPIO_5_OUT_EN位(寄存器0x27的位4)配置以使能输出功能。直通模式也允许RST引脚功能通过R4引脚输出。

当系统处理器已锁定，系统对输入事件无响应时，复位发生信号就很有用。用户可以按下一种复位事件组合以启动系统复位，而无需取出系统电池以执行硬复位。

立即触发时间设置(见表55)仅推荐在噪声非常低、去抖性能良好的情况下使用，否则可能发生误触发。

中断

任一内部中断源有效时， $\overline{\text{INT}}$ 引脚可以置位。用户可以通过寄存器0x3E选择哪些内部中断与外部中断引脚相互作用(见表71)。寄存器0x3D允许用户选择外部中断引脚是保持置位，还是解除置位50 μs 再重新置位，像多个内部中断置位且一个被清除这种情况一样(见表70)。

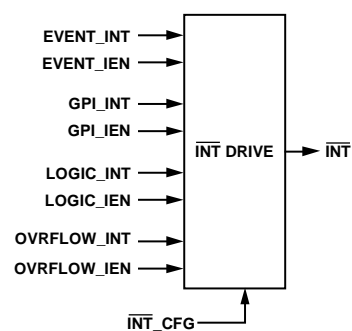


图21. $\overline{\text{INT}}$ 置位低电平

脉冲发生器

ADP5586内置2个脉冲发生器，适合驱动指示器LED驱动信号以及看门狗定时器和其它长时间脉冲应用。所产生脉冲的开启时间和周期可通过8位来定义。为支持长时序，用户可以选择1 ms时钟或125 ms时钟来递增这些定时器。PULSE_GEN_1_PERIOD和PULSE_GEN_2_PERIOD寄存器(寄存器0x30和寄存器0x33)定义这两个脉冲发生器的周期。在PULSE_GEN_CONFIG寄存器(寄存器0x35的位1和位5)中选择125 ms的时钟周期时，脉冲发生器周期最长可以设置为31.875秒。不支持将PULSE_GEN_x_ON_CLK位设置为125 ms的步长，以及将PULSE_GEN_x_PRD_CLK位设置为1 ms的步长。

为支持低电平有效应用，可以通过PULSE_GEN_CONFIG寄存器的位7和位3(PULSE_GEN_x_INV)设置信号反转。可以引入延迟以在通道之间产生同步偏移。如果两个通道同时使能(即通过同一I²C写操作使能)，则延迟之差等于通道之间的偏移。如果仅单个通道有效并且要使延迟同步，则用户首先必须禁用两个脉冲发生器，然后利用同一I²C写命令使能它们。延迟计数器使用与周期计数器相同的时钟选择。更多信息参见表56至表61。要使脉冲发生器通过C1和/或C0输出，必须使能GPIO_8_OUT_EN位和/或GPIO_7_OUT_EN位(寄存器0x28的位[1:0])。

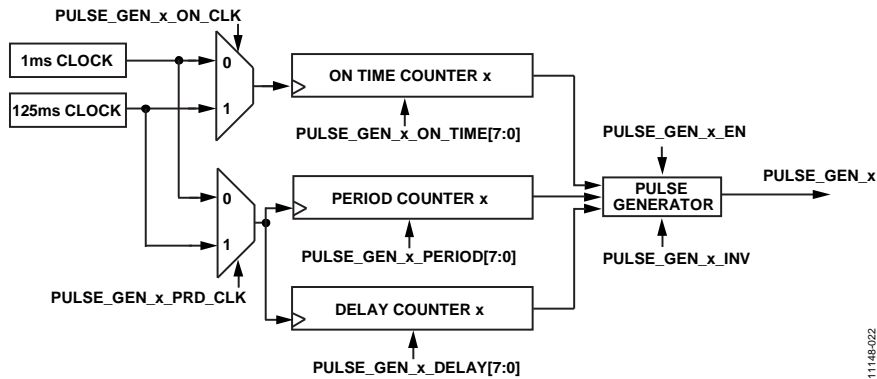


图22. 脉冲发生器框图

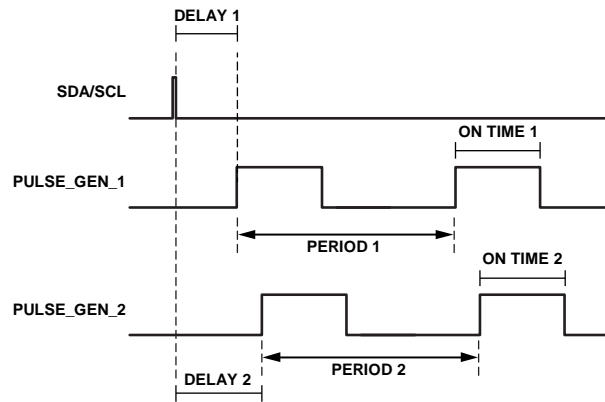


图23. 脉冲发生器时序示例

寄存器接口

通过I²C兼容串行接口可以访问ADP5586的寄存器。该接口可支持高达1 MHz的时钟频率。如果用户正在访问FIFO或键事件计数器(KEC)，FIFO/KEC更新将暂停。如果时钟频率非常低，事件可能不会被及时记录。由于执行I²C读或写操作需要多个I²C周期，因此FIFO或KEC更新可能在中断置位后最长23 μs才发生。此延迟应当不会给用户造成问题。

图24显示了内部寄存器编程的典型写序列。该周期从起始条件开始，接着是硬编码的7位器件地址(ADP5586为0x34)，然后是R/W位，设为0表示写周期。ADP5586通过拉低数据线来应答该地址字节。接下来发送要写入数据的寄存器地址。ADP5586通过拉低数据线来应答该寄存器指针。接下来发送要写入的数据。ADP5586通过拉低数据线来应答该数据字节。最后是停止条件，该序列结束。

图25显示了内部寄存器编程的典型多字节写序列。该周期从起始条件开始，接着是7位器件地址(0x34)，然后是R/W位，设为0表示写周期。ADP5586通过拉低数据线来应答

该地址字节。接下来发送要写入数据的寄存器地址。ADP5586通过拉低数据线来应答该寄存器指针。接下来发送要写入的数据。ADP5586通过拉低数据线来应答该数据字节。然后，指针地址递增以便写入下一数据字节，直到完成数据字节n的写入。每写完一个字节，ADP5586都会拉低数据线，最后通过停止条件结束该序列。

图26显示了读取内部寄存器的典型字节读序列。该周期从起始条件开始，接着是7位器件地址，然后是R/W位，设为0表示写周期。ADP5586通过拉低数据线来应答该地址字节。接下来发送要读取数据的寄存器地址。ADP5586通过拉低数据线来应答该寄存器指针。起始条件重复，接着是7位器件地址(0x34)，然后是R/W位，设为1表示读周期。ADP5586通过拉低数据线来应答该地址字节。然后读取8位数据。主机拉高数据线(不应答)，并通过停止条件结束该序列。



图24. I²C单字节写入序列

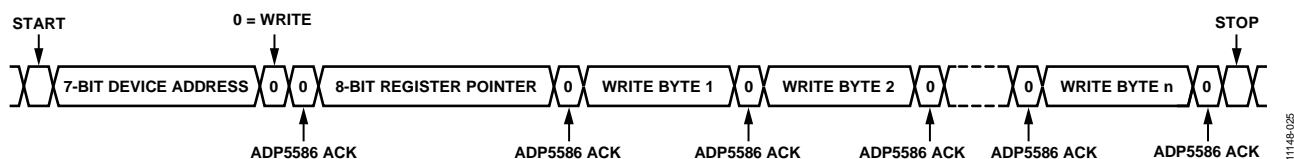


图25. I²C多字节写入序列

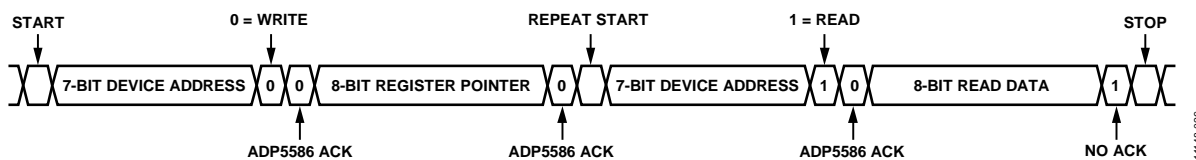


图26. I²C单字节读取序列

ADP5586

图27显示了读取内部寄存器的典型多字节读序列。该周期从起始条件开始，接着是7位器件地址(0x34)，然后是R/W位，设为0表示写周期。ADP5586通过拉低数据线来应答该地址字节。接下来发送要读取数据的寄存器地址。ADP5586通过拉低数据线来应答该寄存器指针。起始条件重复，接着是7位器件地址(0x34)，然后是R/W位，设为1

表示读周期。ADP5586通过拉低数据线来应答该地址字节。接下来读取8位数据。然后，地址指针递增以便读取下一数据字节。对每个字节，主机都会拉低数据线(主机应答)，直到读取数据字节n为止。读取最后一个字节后，主机拉高数据线(不应答)，并通过停止条件结束该序列。

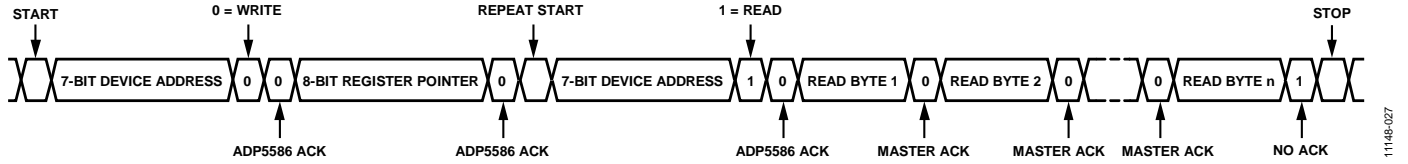


图27. I2C多字节读取序列

11148-027

寄存器映射

表7.

寄存器地址	寄存器名称	R/W ¹	位7	位6	位5	位4	位3	位2	位1	位0	
0x00	ID	R	MAN_ID				REV_ID				
0x01	INT_STATUS	R/W	保留			LOGIC_INT	保留	OVRFLOW_INT	GPI_INT	EVENT_INT	
0x02	Status	R	保留	LOGIC_STAT	保留	EC[4:0]					
0x03	FIFO_1	R	EVENT1_STATE	EVENT1_IDENTIFIER[6:0]							
0x04	FIFO_2	R	EVENT2_STATE	EVENT2_IDENTIFIER[6:0]							
0x05	FIFO_3	R	EVENT3_STATE	EVENT3_IDENTIFIER[6:0]							
0x06	FIFO_4	R	EVENT4_STATE	EVENT4_IDENTIFIER[6:0]							
0x07	FIFO_5	R	EVENT5_STATE	EVENT5_IDENTIFIER[6:0]							
0x08	FIFO_6	R	EVENT6_STATE	EVENT6_IDENTIFIER[6:0]							
0x09	FIFO_7	R	EVENT7_STATE	EVENT7_IDENTIFIER[6:0]							
0x0A	FIFO_8	R	EVENT8_STATE	EVENT8_IDENTIFIER[6:0]							
0x0B	FIFO_9	R	EVENT9_STATE	EVENT9_IDENTIFIER[6:0]							
0x0C	FIFO_10	R	EVENT10_STATE	EVENT10_IDENTIFIER[6:0]							
0x0D	FIFO_11	R	EVENT11_STATE	EVENT11_IDENTIFIER[6:0]							
0x0E	FIFO_12	R	EVENT12_STATE	EVENT12_IDENTIFIER[6:0]							
0x0F	FIFO_13	R	EVENT13_STATE	EVENT13_IDENTIFIER[6:0]							
0x10	FIFO_14	R	EVENT14_STATE	EVENT14_IDENTIFIER[6:0]							
0x11	FIFO_15	R	EVENT15_STATE	EVENT15_IDENTIFIER[6:0]							
0x12	FIFO_16	R	EVENT16_STATE	EVENT16_IDENTIFIER[6:0]							
0x13	GPI_INT_STAT_A	R	保留		GPI_6_INT	GPI_5_INT	GPI_4_INT	GPI_3_INT	GPI_2_INT	GPI_1_INT	
0x14	GPI_INT_STAT_B	R	保留			GPI_11_INT	GPI_10_INT	GPI_9_INT	GPI_8_INT	GPI_7_INT	
0x15	GPI_STATUS_A	R	保留		GPI_6_STAT	GPI_5_STAT	GPI_4_STAT	GPI_3_STAT	GPI_2_STAT	GPI_1_STAT	
0x16	GPI_STATUS_B	R	保留			GPI_11_STAT	GPI_10_STAT	GPI_9_STAT	GPI_8_STAT	GPI_7_STAT	
0x17	R_PULL_CONFIG_A	R/W	R3_PULL_CFG		R2_PULL_CFG		R1_PULL_CFG		R0_PULL_CFG		
0x18	R_PULL_CONFIG_B	R/W	保留				R5_PULL_CFG		R4_PULL_CFG		
0x19	R_PULL_CONFIG_C	R/W	C3_PULL_CFG		C2_PULL_CFG		C1_PULL_CFG		C0_PULL_CFG		
0x1A	R_PULL_CONFIG_D	R/W	保留						C4_PULL_CFG		
0x1B	GPI_INT_LEVEL_A	R/W	保留		GPI_6_INT_LEVEL	GPI_5_INT_LEVEL	GPI_4_INT_LEVEL	GPI_3_INT_LEVEL	GPI_2_INT_LEVEL	GPI_1_INT_LEVEL	
0x1C	GPI_INT_LEVEL_B	R/W	保留			GPI_11_INT_LEVEL	GPI_10_INT_LEVEL	GPI_9_INT_LEVEL	GPI_8_INT_LEVEL	GPI_7_INT_LEVEL	
0x1D	GPI_EVENT_EN_A	R/W	保留		GPI_6_EVENT_EN	GPI_5_EVENT_EN	GPI_4_EVENT_EN	GPI_3_EVENT_EN	GPI_2_EVENT_EN	GPI_1_EVENT_EN	
0x1E	GPI_EVENT_EN_B	R/W	保留			GPI_11_EVENT_EN	GPI_10_EVENT_EN	GPI_9_EVENT_EN	GPI_8_EVENT_EN	GPI_7_EVENT_EN	
0x1F	GPI_INTERRUPT_EN_A	R/W	保留		GPI_6_INT_EN	GPI_5_INT_EN	GPI_4_INT_EN	GPI_3_INT_EN	GPI_2_INT_EN	GPI_1_INT_EN	
0x20	GPI_INTERRUPT_EN_B	R/W	保留			GPI_11_INT_EN	GPI_10_INT_EN	GPI_9_INT_EN	GPI_8_INT_EN	GPI_7_INT_EN	
0x21	DEBOUNCE_DIS_A	R/W	保留		GPI_6_DEB_DIS	GPI_5_DEB_DIS	GPI_4_DEB_DIS	GPI_3_DEB_DIS	GPI_2_DEB_DIS	GPI_1_DEB_DIS	
0x22	DEBOUNCE_DIS_B	R/W	保留			GPI_11_DEB_DIS	GPI_10_DEB_DIS	GPI_9_DEB_DIS	GPI_8_DEB_DIS	GPI_7_DEB_DIS	
0x23	GPO_DATA_OUT_A	R/W	保留		GPO_6_DATA	GPO_5_DATA	GPO_4_DATA	GPO_3_DATA	GPO_2_DATA	GPO_1_DATA	
0x24	GPO_DATA_OUT_B	R/W	保留			GPO_11_DATA	GPO_10_DATA	GPO_9_DATA	GPO_8_DATA	GPO_7_DATA	
0x25	GPO_OUT_MODE_A	R/W	保留		GPO_6_OUT_MODE	GPO_5_OUT_MODE	GPO_4_OUT_MODE	GPO_3_OUT_MODE	GPO_2_OUT_MODE	GPO_1_OUT_MODE	
0x26	GPO_OUT_MODE_B	R/W	保留			GPO_11_OUT_MODE	GPO_10_OUT_MODE	GPO_9_OUT_MODE	GPO_8_OUT_MODE	GPO_7_OUT_MODE	
0x27	GPIO_OUT_EN_A	R/W	保留		GPIO_6_OUT_EN	GPIO_5_OUT_EN	GPIO_4_OUT_EN	GPIO_3_OUT_EN	GPIO_2_OUT_EN	GPIO_1_OUT_EN	
0x28	GPIO_OUT_EN_B	R/W	保留			GPIO_11_OUT_EN	GPIO_10_OUT_EN	GPIO_9_OUT_EN	GPIO_8_OUT_EN	GPIO_7_OUT_EN	

ADP5586

寄存器地址	寄存器名称	R/W ¹	位7	位6	位5	位4	位3	位2	位1	位0
0x29	GPIO_INP_EN_A	R/W	保留		GPIO_6_INP_EN	GPIO_5_INP_EN	GPIO_4_INP_EN	GPIO_3_INP_EN	GPIO_2_INP_EN	GPIO_1_INP_EN
0x2A	GPIO_INP_EN_B	R/W	保留			GPIO_11_INP_EN	GPIO_10_INP_EN	GPIO_9_INP_EN	GPIO_8_INP_EN	GPIO_7_INP_EN
0x2B	RESET_EVENT_A	R/W	RESET_EVENT_A_LEVEL	RESET_EVENT_A, 位[6:0]						
0x2C	RESET_EVENT_B	R/W	RESET_EVENT_B_LEVEL	RESET_EVENT_B, 位[6:0]						
0x2D	RESET_EVENT_C	R/W	RESET_EVENT_C_LEVEL	RESET_EVENT_C, 位[6:0]						
0x2E	RESET_CFG	R/W	RESET_POL	RST_PASSTHRU_EN	RESET_TRIG_TIME, 位[3:0]				RESET_PULSE_WIDTH, 位[1:0]	
0x2F	PULSE_GEN_1_DELAY	R/W	PULSE_GEN_1_DELAY, 位[7:0]							
0x30	PULSE_GEN_1_PERIOD	R/W	PULSE_GEN_1_PERIOD, 位[7:0]							
0x31	PULSE_GEN_1_ON_TIME	R/W	PULSE_GEN_1_ON_TIME, 位[7:0]							
0x32	PULSE_GEN_2_DELAY	R/W	PULSE_GEN_2_DELAY, 位[7:0]							
0x33	PULSE_GEN_2_PERIOD	R/W	PULSE_GEN_2_PERIOD, 位[7:0]							
0x34	PULSE_GEN_2_ON_TIME	R/W	PULSE_GEN_2_ON_TIME, 位[7:0]							
0x35	PULSE_GEN_CONFIG	R/W	PULSE_GEN_1_INV	PULSE_GEN_1_ON_CLK	PULSE_GEN_1_PRD_CLK	PULSE_GEN_1_EN	PULSE_GEN_2_INV	PULSE_GEN_2_ON_CLK	PULSE_GEN_2_PRD_CLK	PULSE_GEN_2_EN
0x36	LOGIC_CFG	R/W	保留	LY_INV	LC_INV	LB_INV	LA_INV	LOGIC_SEL, 位[2:0]		
0x37	LOGIC_FF_CFG	R/W	保留						FF_SET	FF_CLR
0x38	LOGIC_INT_EVENT_EN	R/W	保留					LY_DBNC_DIS	LOGIC_EVENT_EN	LOGIC_INT_LEVEL
0x39	POLL_TIME_CFG	R/W	保留				PRECHARGE_TIME	保留	KEY_POLL_TIME, 位[1:0]	
0x3A	PIN_CONFIG_A	R/W	保留		R5_CONFIG	R4_CONFIG	R3_CONFIG	R2_CONFIG	R1_CONFIG	R0_CONFIG
0x3B	PIN_CONFIG_B	R/W	保留			C4_CONFIG	C3_CONFIG	C2_CONFIG	C1_CONFIG	C0_CONFIG
0x3C	PIN_CONFIG_C	R/W	PULL_SELECT	CO_EXTEND_CFG	R4_EXTEND_CFG	C1_EXTEND_CFG	R3_EXTEND_CFG	保留		R0_EXTEND_CFG
0x3D	GENERAL_CFG	R/W	OSC_EN	OSC_FREQ, 位[1:0]		保留		SW_RESET	INT_CFG	RST_CFG
0x3E	INT_EN	R/W	保留			LOGIC_IEN	保留	OVRFLOW_IEN	GPI_IEN	EVENT_IEN

¹R表示“读”，W表示“写”，R/W表示“读/写”。

寄存器描述

注意：除非另有说明，所有寄存器的默认值为0000 0000。

ID，寄存器0x00

默认值：0011 XXXX(其中X为无关位)

表8. ID的位功能描述

位	位的名称	访问类型	说明
[7:4]	MAN_ID	只读	制造商ID，默认值 = 0011
[3:0]	REV_ID	只读	版本ID

INT_STATUS，寄存器0x01

表9. INT_STATUS的位功能描述

位	位的名称	访问类型	描述 ¹
[7:5]	保留	保留	保留
4	LOGIC_INT	Read/write	0 = 无中断。 1 = 一般逻辑条件引起的中断。
3	保留	保留	保留
2	OVERFLOW_INT	读/写	0 = 无中断。 1 = 溢出条件引起的中断。
1	GPI_INT	读/写	已被配置更新FIFO和事件计数的GPI无法设置此位。 要将此位清0，所有GPI_x_INT位都必须清0。 0 = 无中断。 1 = 一般GPI条件引起的中断。
0	EVENT_INT	读/写	0 = 无中断。 1 = 键事件(按下/释放)、GPI事件(GPI配置为更新FIFO)或逻辑事件(配置为更新FIFO)引起的中断。

¹ 中断位清0需将1写入标志位；写入0或读取标志位不起作用。

Status，寄存器0x02

表10. Status的位功能描述

位	位的名称	访问类型	说明
7	保留	保留	保留。
6	LOGIC_STAT	只读	0 = 逻辑模块(LY)的输出为低电平。 1 = 逻辑模块(LY)的输出为高电平。
5	保留	保留	保留。
[4:0]	EC[4:0]	只读	事件计数值。表示FIFO当前存储了多少事件。

FIFO_1，寄存器0x03

表11. FIFO_1的位功能描述

位	位的名称	访问类型	说明
7	EVENT1_STATE	只读	此位表示EVENT1_IDENTIFIER[6:0]位中记录的事件的状态。 对于事件1到事件36的键事件，使用下列设置： 1 = 键按下。 0 = 键释放。 对于事件37到事件48的GPI和逻辑事件，使用下列设置： 1 = GPI/逻辑有效。 0 = GPI/逻辑无效。 事件37到事件48的有效和无效状态是可编程的。
[6:0]	EVENT1_IDENTIFIER[6:0]	只读	包含该引脚的事件标识符。事件解码信息参见表12。

ADP5586

表12. 事件解码

事件编号	含义	事件编号	含义
0	无事件	25	键25 (R4, C4)
1	键1 (R0, C0)	26	键26 (R5, C0)
2	键2 (R0, C1)	27	键27 (R5, C1)
3	键3 (R0, C2)	28	键28 (R5, C2)
4	键4 (R0, C3)	29	键29 (R5, C3)
5	键5 (R0, C4)	30	键30 (R5, C4)
6	键6 (R1, C0)	31	键31 (R0, GND)
7	键7 (R1, C1)	32	键32 (R1, GND)
8	键8 (R1, C2)	33	键33 (R2, GND)
9	键9 (R1, C3)	34	键34 (R3, GND)
10	键10 (R1, C4)	35	键35 (R4, GND)
11	键11 (R2, C0)	36	键36 (R5, GND)
12	键12 (R2, C1)	37	GPI 1 (R0)
13	键13 (R2, C2)	38	GPI 2 (R1)
14	键14 (R2, C3)	39	GPI 3 (R2)
15	键15 (R2, C4)	40	GPI 4 (R3)
16	键16 (R3, C0)	41	GPI 5 (R4)
17	键17 (R3, C1)	42	GPI 6 (R5)
18	键18 (R3, C2)	43	GPI 7 (C0)
19	键19 (R3, C3)	44	GPI 8 (C1)
20	键20 (R3, C4)	45	GPI 9 (C2)
21	键21 (R4, C0)	46	GPI 10 (C3)
22	键22 (R4, C1)	47	GPI 11 (C4)
23	键23 (R4, C2)	48	逻辑
24	键24 (R4, C3)	49至127	未用

FIFO_2, 寄存器0x04

表13. FIFO_2的位功能描述

位	位的名称	访问类型	说明
7	EVENT2_STATE	只读	位功能描述参见表11。
[6:0]	EVENT2_IDENTIFIER[6:0]	只读	位功能描述参见表11。

FIFO_3, 寄存器0x05

表14. FIFO_3的位功能描述

位	位的名称	访问类型	说明
7	EVENT3_STATE	只读	位功能描述参见表11。
[6:0]	EVENT3_IDENTIFIER[6:0]	只读	位功能描述参见表11。

FIFO_4, 寄存器0x06

表15. FIFO_4的位功能描述

位	位的名称	访问类型	说明
7	EVENT4_STATE	只读	位功能描述参见表11。
[6:0]	EVENT4_IDENTIFIER[6:0]	只读	位功能描述参见表11。

FIFO_5, 寄存器0x07

表16. FIFO_5的位功能描述

位	位的名称	访问类型	说明
7	EVENT5_STATE	只读	位功能描述参见表11。
[6:0]	EVENT5_IDENTIFIER[6:0]	只读	位功能描述参见表11。

FIFO_6, 寄存器0x08**表17. FIFO_6的位功能描述**

位	位的名称	访问类型	说明
7	EVENT6_STATE	只读	位功能描述参见表11。
[6:0]	EVENT6_IDENTIFIER[6:0]	只读	位功能描述参见表11。

FIFO_7, 寄存器0x09**表18. FIFO_7的位功能描述**

位	位的名称	访问类型	说明
7	EVENT7_STATE	只读	位功能描述参见表11。
[6:0]	EVENT7_IDENTIFIER[6:0]	只读	位功能描述参见表11。

FIFO_8, 寄存器0x0A**表19. FIFO_8的位功能描述**

位	位的名称	访问类型	说明
7	EVENT8_STATE	只读	位功能描述参见表11。
[6:0]	EVENT8_IDENTIFIER[6:0]	只读	位功能描述参见表11。

FIFO_9, 寄存器0x0B**表20. FIFO_9的位功能描述**

位	位的名称	访问类型	说明
7	EVENT9_STATE	只读	位功能描述参见表11。
[6:0]	EVENT9_IDENTIFIER[6:0]	只读	位功能描述参见表11。

FIFO_10, 寄存器0x0C**表21. FIFO_10的位功能描述**

位	位的名称	访问类型	说明
7	EVENT10_STATE	只读	位功能描述参见表11。
[6:0]	EVENT10_IDENTIFIER[6:0]	只读	位功能描述参见表11。

FIFO_11, 寄存器0x0D**表22. FIFO_11的位功能描述**

位	位的名称	访问类型	说明
7	EVENT11_STATE	只读	位功能描述参见表11。
[6:0]	EVENT11_IDENTIFIER[6:0]	只读	位功能描述参见表11。

FIFO_12, 寄存器0x0E**表23. FIFO_12的位功能描述**

位	位的名称	访问类型	说明
7	EVENT12_STATE	只读	位功能描述参见表11。
[6:0]	EVENT12_IDENTIFIER[6:0]	只读	位功能描述参见表11。

FIFO_13, 寄存器0x0F**表24. FIFO_13的位功能描述**

位	位的名称	访问类型	说明
7	EVENT13_STATE	只读	位功能描述参见表11。
[6:0]	EVENT13_IDENTIFIER[6:0]	只读	位功能描述参见表11。

ADP5586

FIFO_14, 寄存器0x10

表25. FIFO_14的位功能描述

位	位的名称	访问类型	说明
7	EVENT14_STATE	只读	位功能描述参见表11。
[6:0]	EVENT14_IDENTIFIER[6:0]	只读	位功能描述参见表11。

FIFO_15, 寄存器0x11

表26. FIFO_15的位功能描述

位	位的名称	访问类型	说明
7	EVENT15_STATE	只读	位功能描述参见表11。
[6:0]	EVENT15_IDENTIFIER[6:0]	只读	位功能描述参见表11。

FIFO_16, 寄存器0x12

表27. FIFO_16的位功能描述

位	位的名称	访问类型	说明
7	EVENT16_STATE	只读	位功能描述参见表11。
[6:0]	EVENT16_IDENTIFIER[6:0]	只读	位功能描述参见表11。

GPI_INT_STAT_A, 寄存器0x13

表28. GPI_INT_STAT_A的位功能描述

位	位的名称	访问类型	说明
[7:6]	保留	保留	保留。
5	GPI_6_INT	只读	0 = 无中断 1 = GPI 6(R5引脚)引起的中断。读取清0。
4	GPI_5_INT	只读	0 = 无中断 1 = GPI 5(R4引脚)引起的中断。读取清0。
3	GPI_4_INT	只读	0 = 无中断 1 = GPI 4(R3引脚)引起的中断。读取清0。
2	GPI_3_INT	只读	0 = 无中断 1 = GPI 3(R2引脚)引起的中断。读取清0。
1	GPI_2_INT	只读	0 = 无中断 1 = GPI 2(R1引脚)引起的中断。读取清0。
0	GPI_1_INT	只读	0 = 无中断 1 = GPI 1(R0引脚)引起的中断。读取清0。

GPI_INT_STAT_B, 寄存器0x14

表29. GPI_INT_STAT_B的位功能描述

位	位的名称	访问类型	说明
[7:5]	保留	保留	保留。
4	GPI_11_INT	只读	0 = 无中断。 1 = GPI 11(C4引脚)引起的中断。读取清0。
3	GPI_10_INT	只读	0 = 无中断。 1 = GPI 10(C3引脚)引起的中断。读取清0。
2	GPI_9_INT	只读	0 = 无中断。 1 = GPI 9(C2引脚)引起的中断。读取清0。
1	GPI_8_INT	只读	0 = 无中断。 1 = GPI 8(C1引脚)引起的中断。读取清0。
0	GPI_7_INT	只读	0 = 无中断。 1 = GPI 7(C0引脚)引起的中断。读取清0。

GPI_STATUS_A, 寄存器0x15**表30. GPI_STATUS_A的位功能描述**

位	位的名称	访问类型	说明
[7:6]	保留	保留	保留.
5	GPI_6_STAT	只读	0 = GPI 6(R5引脚)为低电平。 1 = GPI 6(R5引脚)为高电平。
4	GPI_5_STAT	只读	0 = GPI 5(R4引脚)为低电平。 1 = GPI 5(R4引脚)为高电平。
3	GPI_4_STAT	只读	0 = GPI 4(R3引脚)为低电平。 1 = GPI 4(R3引脚)为高电平。
2	GPI_3_STAT	只读	0 = GPI 3(R2引脚)为低电平。 1 = GPI 3(R2引脚)为高电平。
1	GPI_2_STAT	只读	0 = GPI 2(R1引脚)为低电平。 1 = GPI 2(R1引脚)为高电平。
0	GPI_1_STAT	只读	0 = GPI 1(R0引脚)为低电平。 1 = GPI 1(R0引脚)为高电平。

GPI_STATUS_B, 寄存器0x16**表31. GPI_STATUS_B的位功能描述**

位	位的名称	访问类型	说明
[7:5]	保留	保留	保留.
4	GPI_11_STAT	只读	0 = GPI 11(C4引脚)为低电平。 1 = GPI 11(C4引脚)为高电平。
3	GPI_10_STAT	只读	0 = GPI 10(C3引脚)为低电平。 1 = GPI 10(C3引脚)为高电平。
2	GPI_9_STAT	只读	0 = GPI 9(C2引脚)为低电平。 1 = GPI 9(C2引脚)为高电平。
1	GPI_8_STAT	只读	0 = GPI 8(C1引脚)为低电平。 1 = GPI 8(C1引脚)为高电平。
0	GPI_7_STAT	只读	0 = GPI 7(C0引脚)为低电平。 1 = GPI 7(C0引脚)为高电平。

ADP5586

R_PULL_CONFIG_A, 寄存器0x17

默认值 = 0101 0101

表32. R_PULL_CONFIG_A的位功能描述

位	位的名称	访问类型	说明
[7:6]	R3_PULL_CFG	读/写	00 = 使能300 kΩ上拉电阻。 01 = 使能300 kΩ下拉电阻。 10 = 使能100 kΩ上拉电阻。 11 = 禁用所有上拉和下拉电阻。
[5:4]	R2_PULL_CFG	读/写	00 = 使能300 kΩ上拉电阻。 01 = 使能300 kΩ下拉电阻。 10 = 使能100 kΩ上拉电阻。 11 = 禁用所有上拉和下拉电阻。
[3:2]	R1_PULL_CFG	读/写	00 = 使能300 kΩ上拉电阻。 01 = 使能300 kΩ下拉电阻。 10 = 使能100 kΩ上拉电阻。 11 = 禁用所有上拉和下拉电阻。
[1:0]	R0_PULL_CFG	读/写	00 = 使能300 kΩ上拉电阻。 01 = 使能300 kΩ下拉电阻。 10 = 使能100 kΩ上拉电阻。 11 = 禁用所有上拉和下拉电阻。

R_PULL_CONFIG_B, 寄存器0x18

默认值 = 0000 0101

表33. R_PULL_CONFIG_B的位功能描述

位	位的名称	访问类型	说明
[7:4]	保留	保留	保留。
[3:2]	R5_PULL_CFG	读/写	保留, ADP5586ACBZ-01-R7选项除外。 00 = 使能300 kΩ上拉电阻。 01 = 使能300 kΩ下拉电阻。 10 = 使能100 kΩ上拉电阻。 11 = 禁用所有上拉和下拉电阻。
[1:0]	R4_PULL_CFG	读/写	00 = 使能300 kΩ上拉电阻。 01 = 使能300 kΩ下拉电阻。 10 = 使能100 kΩ上拉电阻。 11 = 禁用所有上拉和下拉电阻。

R_PULL_CONFIG_C, 寄存器0x19

默认值 = 0101 0001

表34. R_PULL_CONFIG_C的位功能描述

位	位的名称	访问类型	说明
[7:6]	C3_PULL_CFG	读/写	00 = 使能300 kΩ上拉电阻。 01 = 使能300 kΩ下拉电阻。 10 = 使能100 kΩ上拉电阻。 11 = 禁用所有上拉和下拉电阻。
[5:4]	C2_PULL_CFG	读/写	00 = 使能300 kΩ上拉电阻。 01 = 使能300 kΩ下拉电阻。 10 = 使能100 kΩ上拉电阻。 11 = 禁用所有上拉和下拉电阻。
[3:2]	C1_PULL_CFG	读/写	00 = 使能300 kΩ上拉电阻。 01 = 使能300 kΩ下拉电阻。 10 = 使能100 kΩ上拉电阻。 11 = 禁用所有上拉和下拉电阻。
[1:0]	C0_PULL_CFG	读/写	00 = 使能300 kΩ上拉电阻。 01 = 使能300 kΩ下拉电阻。 10 = 使能100 kΩ上拉电阻。 11 = 禁用所有上拉和下拉电阻。

R_PULL_CONFIG_D, 寄存器0x1A

默认值 = 0000 0001

表35. R_PULL_CONFIG_D的位功能描述

位	位的名称	访问类型	说明
[7:2]	保留	保留	保留.
[1:0]	C4_PULL_CFG	读/写	00 = 使能300 kΩ上拉电阻。 01 = 使能300 kΩ下拉电阻。 10 = 使能100 kΩ上拉电阻。 11 = 禁用所有上拉和下拉电阻。

GPI_INT_LEVEL_A, 寄存器0x1B**表36. GPI_INT_LEVEL_A的位功能描述**

位	位的名称	访问类型	说明
[7:6]	保留	保留	保留.
5	GPI_6_INT_LEVEL	读/写	0 = GPI 6中断低电平有效(只要R5为低电平, GPI_6_INT即置1)。 1 = GPI 6中断高电平有效(只要R5为高电平, GPI_6_INT即置1)。
4	GPI_5_INT_LEVEL	读/写	0 = GPI 5中断低电平有效(只要R4为低电平, GPI_5_INT即置1)。 1 = GPI 5中断高电平有效(只要R4为高电平, GPI_5_INT即置1)。
3	GPI_4_INT_LEVEL	读/写	0 = GPI 4中断低电平有效(只要R3为低电平, GPI_4_INT即置1)。 1 = GPI 4中断高电平有效(只要R3为高电平, GPI_4_INT即置1)。
2	GPI_3_INT_LEVEL	读/写	0 = GPI 3中断低电平有效(只要R2为低电平, GPI_3_INT即置1)。 1 = GPI 3中断高电平有效(只要R2为高电平, GPI_3_INT即置1)。
1	GPI_2_INT_LEVEL	读/写	0 = GPI 2中断低电平有效(只要R1为低电平, GPI_2_INT即置1)。 1 = GPI 2中断高电平有效(只要R1为高电平, GPI_2_INT即置1)。
0	GPI_1_INT_LEVEL	读/写	0 = GPI 1中断低电平有效(只要R0为低电平, GPI_1_INT即置1)。 1 = GPI 1中断高电平有效(只要R0为高电平, GPI_1_INT即置1)。

GPI_INT_LEVEL_B, 寄存器0x1C**表37. GPI_INT_LEVEL_B的位功能描述**

位	位的名称	访问类型	说明
[7:5]	保留	保留	保留.
4	GPI_11_INT_LEVEL	读/写	0 = GPI 11中断低电平有效(只要R10为低电平, GPI_11_INT即置1)。 1 = GPI 11中断高电平有效(只要R10为高电平, GPI_11_INT即置1)。
3	GPI_10_INT_LEVEL	读/写	0 = GPI 10中断低电平有效(只要R9为低电平, GPI_10_INT即置1)。 1 = GPI 10中断高电平有效(只要R9为高电平, GPI_10_INT即置1)。
2	GPI_9_INT_LEVEL	读/写	0 = GPI 9中断低电平有效(只要R8为低电平, GPI_9_INT即置1)。 1 = GPI 9中断高电平有效(只要R8为高电平, GPI_9_INT即置1)。
1	GPI_8_INT_LEVEL	读/写	0 = GPI 8中断低电平有效(只要R7为低电平, GPI_8_INT即置1)。 1 = GPI 8中断高电平有效(只要R7为高电平, GPI_8_INT即置1)。
0	GPI_7_INT_LEVEL	读/写	0 = GPI 7中断低电平有效(只要R6为低电平, GPI_7_INT即置1)。 1 = GPI 7中断高电平有效(只要R6为高电平, GPI_7_INT即置1)。

ADP5586

GPI_EVENT_EN_A, 寄存器0x1D

表38. GPI_EVENT_EN_A的位功能描述

位	位的名称	访问类型	说明
[7:6]	保留	保留	保留。
5	GPI_6_EVENT_EN	读/写	0 = 禁用来自GPI 6的GPI事件。 1 = 允许GPI 6活动产生FIFO事件。 ¹
4	GPI_5_EVENT_EN	读/写	0 = 禁用来自GPI 5的GPI事件。 1 = 允许GPI 5活动产生FIFO事件。 ¹
3	GPI_4_EVENT_EN	读/写	0 = 禁用来自GPI 4的GPI事件。 1 = 允许GPI 4活动产生FIFO事件。 ¹
2	GPI_3_EVENT_EN	读/写	0 = 禁用来自GPI 3的GPI事件。 1 = 允许GPI 3活动产生FIFO事件。 ¹
1	GPI_2_EVENT_EN	读/写	0 = 禁用来自GPI 2的GPI事件。 1 = 允许GPI 2活动产生FIFO事件。 ¹
0	GPI_1_EVENT_EN	读/写	0 = 禁用来自GPI 1的GPI事件。 1 = 允许GPI 1活动产生FIFO事件。 ¹

¹ 这种模式下的GPI被认为是FIFO事件，可用于解锁。这种模式下的GPI活动引起EVENT_INT中断。这种模式下，GPI不产生GPI_INT中断。

GPI_EVENT_EN_B, 寄存器0x1E

表39. GPI_EVENT_EN_B的位功能描述

位	位的名称	访问类型	说明
[7:5]	保留	保留	保留。
4	GPI_11_EVENT_EN	读/写	0 = 禁用来自GPI 11的GPI事件。 1 = 允许GPI 11活动产生FIFO事件。 ¹
3	GPI_10_EVENT_EN	读/写	0 = 禁用来自GPI 10的GPI事件。 1 = 允许GPI 10活动产生FIFO事件。 ¹
2	GPI_9_EVENT_EN	读/写	0 = 禁用来自GPI 9的GPI事件。 1 = 允许GPI 9活动产生FIFO事件。 ¹
1	GPI_8_EVENT_EN	读/写	0 = 禁用来自GPI 8的GPI事件。 1 = 允许GPI 8活动产生FIFO事件。 ¹
0	GPI_7_EVENT_EN	读/写	0 = 禁用来自GPI 7的GPI事件。 1 = 允许GPI 7活动产生FIFO事件。 ¹

¹ 这种模式下的GPI被认为是FIFO事件，可用于解锁。这种模式下的GPI活动引起EVENT_INT中断。这种模式下，GPI不产生GPI_INT中断。

GPI_INTERRUPT_EN_A, 寄存器0x1F**表40. GPI_INTERRUPT_EN_A的位功能描述**

位	位的名称	访问类型	说明
[7:6]	保留	保留	保留。
5	GPI_6_INT_EN	读/写	0 = 禁用GPI_6_INT。 1 = 使能GPI_6_INT。如果GPI_6_INT置1且满足GPI 6中断条件，则GPI_INT位(寄存器0x01的位1)置位。
4	GPI_5_INT_EN	读/写	0 = 禁用GPI_5_INT。 1 = 使能GPI_5_INT。如果GPI_5_INT置1且满足GPI 5中断条件，则GPI_INT位(寄存器0x01的位1)置位。
3	GPI_4_INT_EN	读/写	0 = 禁用GPI_4_INT。 1 = 使能GPI_4_INT。如果GPI_4_INT置1且满足GPI 4中断条件，则GPI_INT位(寄存器0x01的位1)置位。
2	GPI_3_INT_EN	读/写	0 = 禁用GPI_3_INT。 1 = 使能GPI_3_INT。如果GPI_3_INT置1且满足GPI 3中断条件，则GPI_INT位(寄存器0x01的位1)置位。
1	GPI_2_INT_EN	读/写	0 = 禁用GPI_2_INT。 1 = 使能GPI_2_INT。如果GPI_2_INT置1且满足GPI 2中断条件，则GPI_INT位(寄存器0x01的位1)置位。
0	GPI_1_INT_EN	读/写	0 = 禁用GPI_1_INT。 1 = 使能GPI_1_INT。如果GPI_1_INT置1且满足GPI 1中断条件，则GPI_INT位(寄存器0x01的位1)置位。

GPI_INTERRUPT_EN_B, 寄存器0x20**表41. GPI_INTERRUPT_EN_B的位功能描述**

位	位的名称	访问类型	说明
[7:5]	保留	保留	保留。
4	GPI_11_INT_EN	读/写	0 = 禁用GPI_11_INT。 1 = 使能GPI_11_INT。如果GPI_11_INT置1且满足GPI 11中断条件，则GPI_INT位(寄存器0x01的位1)置位。
3	GPI_10_INT_EN	读/写	0 = 禁用GPI_10_INT。 1 = 使能GPI_10_INT。如果GPI_10_INT置1且满足GPI 10中断条件，则GPI_INT位(寄存器0x01的位1)置位。
2	GPI_9_INT_EN	读/写	0 = 禁用GPI_9_INT。 1 = 使能GPI_9_INT。如果GPI_9_INT置1且满足GPI 9中断条件，则GPI_INT位(寄存器0x01的位1)置位。
1	GPI_8_INT_EN	读/写	0 = 禁用GPI_8_INT。 1 = 使能GPI_8_INT。如果GPI_8_INT置1且满足GPI 8中断条件，则GPI_INT位(寄存器0x01的位1)置位。
0	GPI_7_INT_EN	读/写	0 = 禁用GPI_7_INT。 1 = 使能GPI_7_INT。如果GPI_7_INT置1且满足GPI 7中断条件，则GPI_INT位(寄存器0x01的位1)置位。

ADP5586

DEBOUNCE_DIS_A, 寄存器0x21

表42. DEBOUNCE_DIS_A的位功能描述

位	位的名称	访问类型	说明
[7:6]	保留	保留	保留.
5	GPI_6_DEB_DIS	读/写	0 = 使能GPI 6上的去抖。 1 = 禁用GPI 6上的去抖。
4	GPI_5_DEB_DIS	读/写	0 = 使能GPI 5上的去抖。 1 = 禁用GPI 5上的去抖。
3	GPI_4_DEB_DIS	读/写	0 = 使能GPI 4上的去抖。 1 = 禁用GPI 4上的去抖。
2	GPI_3_DEB_DIS	读/写	0 = 使能GPI 3上的去抖。 1 = 禁用GPI 3上的去抖。
1	GPI_2_DEB_DIS	读/写	0 = 使能GPI 2上的去抖。 1 = 禁用GPI 2上的去抖。
0	GPI_1_DEB_DIS	读/写	0 = 使能GPI 1上的去抖。 1 = 禁用GPI 1上的去抖。

DEBOUNCE_DIS_B, 寄存器0x22

表43. DEBOUNCE_DIS_B的位功能描述

位	位的名称	访问类型	说明
[7:5]	保留	保留	保留.
4	GPI_11_DEB_DIS	读/写	0 = 使能GPI 11上的去抖。 1 = 禁用GPI 11上的去抖。
3	GPI_10_DEB_DIS	读/写	0 = 使能GPI 10上的去抖。 1 = 禁用GPI 10上的去抖。
2	GPI_9_DEB_DIS	读/写	0 = 使能GPI 9上的去抖。 1 = 禁用GPI 9上的去抖。
1	GPI_8_DEB_DIS	读/写	0 = 使能GPI 8上的去抖。 1 = 禁用GPI 8上的去抖。
0	GPI_7_DEB_DIS	读/写	0 = 使能GPI 7上的去抖。 1 = 禁用GPI 7上的去抖。

GPO_DATA_OUT_A, 寄存器0x23

表44. GPO_DATA_OUT_A的位功能描述

位	位的名称	访问类型	说明
[7:6]	保留	保留	保留.
5	GPO_6_DATA	读/写	0 = 设置输出低电平。 1 = 设置输出高电平。
4	GPO_5_DATA	读/写	0 = 设置输出低电平。 1 = 设置输出高电平。
3	GPO_4_DATA	读/写	0 = 设置输出低电平。 1 = 设置输出高电平。
2	GPO_3_DATA	读/写	0 = 设置输出低电平。 1 = 设置输出高电平。
1	GPO_2_DATA	读/写	0 = 设置输出低电平。 1 = 设置输出高电平。
0	GPO_1_DATA	读/写	0 = 设置输出低电平。 1 = 设置输出高电平。

GPO_DATA_OUT_B, 寄存器0x24**表45. GPO_DATA_OUT_B的位功能描述**

位	位的名称	访问类型	说明
[7:5]	保留	保留	保留.
4	GPO_11_DATA	读/写	0 = 设置输出低电平。 1 = 设置输出高电平。
3	GPO_10_DATA	读/写	0 = 设置输出低电平。 1 = 设置输出高电平。
2	GPO_9_DATA	读/写	0 = 设置输出低电平。 1 = 设置输出高电平。
1	GPO_8_DATA	读/写	0 = 设置输出低电平。 1 = 设置输出高电平。
0	GPO_7_DATA	读/写	0 = 设置输出低电平。 1 = 设置输出高电平。

GPO_OUT_MODE_A, 寄存器0x25**表46. GPO_OUT_MODE_A的位功能描述**

位	位的名称	访问类型	说明
[7:6]	保留	保留	保留.
5	GPO_6_OUT_MODE	读/写	0 = 推挽。 1 = 开漏。
4	GPO_5_OUT_MODE	读/写	0 = 推挽。 1 = 开漏。
3	GPO_4_OUT_MODE	读/写	0 = 推挽。 1 = 开漏。
2	GPO_3_OUT_MODE	读/写	0 = 推挽。 1 = 开漏。
1	GPO_2_OUT_MODE	读/写	0 = 推挽。 1 = 开漏。
0	GPO_1_OUT_MODE	读/写	0 = 推挽。 1 = 开漏。

GPO_OUT_MODE_B, 寄存器0x26**表47. GPO_OUT_MODE_B的位功能描述**

位	位的名称	访问类型	说明
[7:5]	保留	保留	保留.
4	GPO_11_OUT_MODE	读/写	0 = 推挽。 1 = 开漏。
3	GPO_10_OUT_MODE	读/写	0 = 推挽。 1 = 开漏。
2	GPO_9_OUT_MODE	读/写	0 = 推挽。 1 = 开漏。
1	GPO_8_OUT_MODE	读/写	0 = 推挽。 1 = 开漏。
0	GPO_7_OUT_MODE	读/写	0 = 推挽。 1 = 开漏。

ADP5586

GPIO_OUT_EN_A, 寄存器0x27

表48. GPIO_OUT_EN_A的位功能描述

位	位的名称	访问类型	说明
[7:6]	保留	保留	保留.
5	GPIO_6_OUT_EN	读/写	0 = GPIO 6输出禁用。 1 = GPIO 6输出使能。
4	GPIO_5_OUT_EN	读/写	0 = GPIO 5输出禁用。 1 = GPIO 5输出使能。
3	GPIO_4_OUT_EN	读/写	0 = GPIO 4输出禁用。 1 = GPIO 4输出使能。
2	GPIO_3_OUT_EN	读/写	0 = GPIO 3输出禁用。 1 = GPIO 3输出使能。
1	GPIO_2_OUT_EN	读/写	0 = GPIO 2输出禁用。 1 = GPIO 2输出使能。
0	GPIO_1_OUT_EN	读/写	0 = GPIO 1输出禁用。 1 = GPIO 1输出使能。

GPIO_OUT_EN_B, 寄存器0x28

表49. GPIO_OUT_EN_B的位功能描述

位	位的名称	访问类型	说明
[7:5]	保留	保留	保留.
4	GPIO_11_OUT_EN	读/写	0 = GPIO 11输出禁用。 1 = GPIO 11输出使能。
3	GPIO_10_OUT_EN	读/写	0 = GPIO 10输出禁用。 1 = GPIO 10输出使能。
2	GPIO_9_OUT_EN	读/写	0 = GPIO 9输出禁用。 1 = GPIO 9输出使能。
1	GPIO_8_OUT_EN	读/写	0 = GPIO 8输出禁用。 1 = GPIO 8输出使能。
0	GPIO_7_OUT_EN	读/写	0 = GPIO 7输出禁用。 1 = GPIO 7输出使能。

GPIO_INP_EN_A, 寄存器0x29

表50. GPIO_INP_EN_A的位功能描述

位	位的名称	访问类型	说明
[7:6]	保留	保留	保留.
5	GPIO_6_INP_EN	读/写	0 = GPIO 6输入禁用。 1 = GPIO 6输入使能。
4	GPIO_5_INP_EN	读/写	0 = GPIO 5输入禁用。 1 = GPIO 5输入使能。
3	GPIO_4_INP_EN	读/写	0 = GPIO 4输入禁用。 1 = GPIO 4输入使能。
2	GPIO_3_INP_EN	读/写	0 = GPIO 3输入禁用。 1 = GPIO 3输入使能。
1	GPIO_2_INP_EN	读/写	0 = GPIO 2输入禁用。 1 = GPIO 2输入使能。
0	GPIO_1_INP_EN	读/写	0 = GPIO 1输入禁用。 1 = GPIO 1输入使能。

GPIO_INP_EN_B, 寄存器0x2A**表51. GPIO_INP_EN_B的位功能描述**

位	位的名称	访问类型	说明
[7:5]	保留	保留	保留。
4	GPIO_11_INP_EN	读/写	0 = GPIO 11输入禁用。 1 = GPIO 11输入使能。
3	GPIO_10_INP_EN	读/写	0 = GPIO 10输入禁用。 1 = GPIO 10输入使能。
2	GPIO_9_INP_EN	读/写	0 = GPIO 9输入禁用。 1 = GPIO 9输入使能。
1	GPIO_8_INP_EN	读/写	0 = GPIO 8输入禁用。 1 = GPIO 8输入使能。
0	GPIO_7_INP_EN	读/写	0 = GPIO 7输入禁用。 1 = GPIO 7输入使能。

RESET_EVENT_A, 寄存器0x2B**表52. RESET_EVENT_A的位功能描述**

位	位的名称	访问类型	说明
7	RESET_EVENT_A_LEVEL	读/写	定义第一个复位事件应处于何种水平才能产生RESET_OUT信号。 对于键事件，使用下列设置： 0 = 无效事件用作复位条件。 1 = 有效事件用作复位条件。 对于配置为更新FIFO的GPI和逻辑输出，使用下列设置： 0 = 不适用；释放不用于产生复位。 1 = 按下用作复位事件。
[6:0]	RESET_EVENT_A[6:0]	读/写	定义用来产生RESET_OUT信号的事件。使用RESET_EVENT_A[6:0]、RESET_EVENT_B[6:0]和RESET_EVENT_C[6:0]，最多可以定义三个事件来产生RESET_OUT信号。如果其中一个寄存器为0，则该寄存器不用于产生复位。必须同时检测到所有复位事件才能触发复位。

RESET_EVENT_B, 寄存器0x2C**表53. RESET_EVENT_B的位功能描述**

位	位的名称	访问类型	说明
7	RESET_EVENT_B_LEVEL	读/写	定义第二个复位事件应处于何种水平才能产生RESET_OUT信号。见表52。
[6:0]	RESET_EVENT_B[6:0]	读/写	定义用来产生RESET_OUT信号的事件。参见表12。

RESET_EVENT_C, 寄存器0x2D**表54. RESET_EVENT_C的位功能描述**

位	位的名称	访问类型	说明
7	RESET_EVENT_C_LEVEL	读/写	定义第三个复位事件应处于何种水平才能产生RESET_OUT信号。见表52。
[6:0]	RESET_EVENT_C[6:0]	读/写	定义用来产生RESET_OUT信号的事件。参见表12。

ADP5586

RESET_CFG, 寄存器0x2E

表55. RESET_CFG的位功能描述

位	位的名称	访问类型	说明
7	RESET_POL	读/写	设置RESET_OUT信号的极性。 0 = RESET_OUT低电平有效。 1 = RESET_OUT高电平有效。
6	RST_PASSTHRU_EN	读/写	允许RST引脚覆盖(“或”运算)RESET_OUT信号。
[5:2]	RESET_TRIG_TIME[3:0]	读/写	定义复位事件必须保持有效多长时间才能产生RESET_OUT信号。 所有事件必须同时有效并保持同样的时间。 0000 = 立即。 0001 = 1.0 s。 0010 = 1.5 s。 0011 = 2.0 s。 0100 = 2.5 s。 0101 = 3.0 s。 0110 = 3.5 s。 0111 = 4.0 s。 1000 = 5.0 s。 1001 = 6.0 s。 1010 = 7.0 s。 1011 = 8.0 s。 1100 = 9.0 s。 1101 = 10.0 s。 1110 = 11.0 s。 1111 = 12.0 s。
[1:0]	RESET_PULSE_WIDTH[1:0]	读/写	定义RESET_OUT信号的脉冲宽度。 00 = 500 μs。 01 = 1 ms。 10 = 2 ms。 11 = 10 ms。

PULSE_GEN_1_DELAY, 寄存器0x2F

表56. PULSE_GEN_1_DELAY的位功能描述

位	位的名称	访问类型	说明																													
[7:0]	PULSE_GEN_1_DELAY[7:0]	读/写	定义脉冲发生器1的第一个使能的第一个时钟的初始延迟时间。 延迟表示为所选周期时钟速度的时钟周期数(参见寄存器0x35)。 例如:																													
			<table border="1"> <thead> <tr> <th rowspan="2">PULSE_GEN_1_DELAY</th> <th colspan="2">PULSE_GEN_1_PRD_CLK</th> </tr> <tr> <th>0</th> <th>1</th> </tr> </thead> <tbody> <tr> <td>0000 0000</td> <td>0</td> <td>0 ms</td> </tr> <tr> <td>0000 0001</td> <td>1</td> <td>125 ms</td> </tr> <tr> <td>0000 0010</td> <td>2</td> <td>250 ms</td> </tr> <tr> <td>0000 0011</td> <td>3</td> <td>375 ms</td> </tr> <tr> <td>0000 0100</td> <td>4</td> <td>500 ms</td> </tr> <tr> <td>...</td> <td>...</td> <td>...</td> </tr> <tr> <td>1111 1110</td> <td>254 ms</td> <td>31.750 sec</td> </tr> <tr> <td>1111 1111</td> <td>255 ms</td> <td>31.875 sec</td> </tr> </tbody> </table>	PULSE_GEN_1_DELAY	PULSE_GEN_1_PRD_CLK		0	1	0000 0000	0	0 ms	0000 0001	1	125 ms	0000 0010	2	250 ms	0000 0011	3	375 ms	0000 0100	4	500 ms	1111 1110	254 ms	31.750 sec	1111 1111	255 ms	31.875 sec
PULSE_GEN_1_DELAY	PULSE_GEN_1_PRD_CLK																															
	0	1																														
0000 0000	0	0 ms																														
0000 0001	1	125 ms																														
0000 0010	2	250 ms																														
0000 0011	3	375 ms																														
0000 0100	4	500 ms																														
...																														
1111 1110	254 ms	31.750 sec																														
1111 1111	255 ms	31.875 sec																														

PULSE_GEN_1_PERIOD, 寄存器0x30**表57. PULSE_GEN_1_PERIOD的位功能描述**

位	位的名称	访问类型	说明																													
[7:0]	PULSE_GEN_1_PERIOD[7:0]	读/写	定义脉冲发生器1的周期。周期表示为所选周期时钟速度的时钟周期数(参见寄存器0x35)。例如:																													
			<table border="1"> <thead> <tr> <th rowspan="2">PULSE_GEN_1_PERIOD</th> <th colspan="2">PULSE_GEN_1_PRD_CLK</th> </tr> <tr> <th>0</th> <th>1</th> </tr> </thead> <tbody> <tr> <td>0000 0000</td> <td>0 ms</td> <td>0 ms</td> </tr> <tr> <td>0000 0001</td> <td>1 ms</td> <td>125 ms</td> </tr> <tr> <td>0000 0010</td> <td>2 ms</td> <td>250 ms</td> </tr> <tr> <td>0000 0011</td> <td>3 ms</td> <td>375 ms</td> </tr> <tr> <td>0000 0100</td> <td>4 ms</td> <td>500 ms</td> </tr> <tr> <td>...</td> <td>...</td> <td>...</td> </tr> <tr> <td>1111 1110</td> <td>254 ms</td> <td>31.750 sec</td> </tr> <tr> <td>1111 1111</td> <td>255 ms</td> <td>31.875 sec</td> </tr> </tbody> </table>	PULSE_GEN_1_PERIOD	PULSE_GEN_1_PRD_CLK		0	1	0000 0000	0 ms	0 ms	0000 0001	1 ms	125 ms	0000 0010	2 ms	250 ms	0000 0011	3 ms	375 ms	0000 0100	4 ms	500 ms	1111 1110	254 ms	31.750 sec	1111 1111	255 ms	31.875 sec
PULSE_GEN_1_PERIOD	PULSE_GEN_1_PRD_CLK																															
	0	1																														
0000 0000	0 ms	0 ms																														
0000 0001	1 ms	125 ms																														
0000 0010	2 ms	250 ms																														
0000 0011	3 ms	375 ms																														
0000 0100	4 ms	500 ms																														
...																														
1111 1110	254 ms	31.750 sec																														
1111 1111	255 ms	31.875 sec																														

PULSE_GEN_1_ON_TIME, 寄存器0x31**表58. PULSE_GEN_1_ON_TIME的位功能描述**

位	位的名称	访问类型	说明																													
[7:0]	PULSE_GEN_1_ON_TIME[7:0]	读/写	定义脉冲发生器1的开启时间。开启时间表示为所选时钟速度的时钟周期数(参见寄存器0x35)。例如:																													
			<table border="1"> <thead> <tr> <th rowspan="2">PULSE_GEN_1_ON_TIME</th> <th colspan="2">PULSE_GEN_1_ON_CLK</th> </tr> <tr> <th>0</th> <th>1</th> </tr> </thead> <tbody> <tr> <td>0000 0000</td> <td>0 ms</td> <td>0 ms</td> </tr> <tr> <td>0000 0001</td> <td>1 ms</td> <td>125 ms</td> </tr> <tr> <td>0000 0010</td> <td>2 ms</td> <td>250 ms</td> </tr> <tr> <td>0000 0011</td> <td>3 ms</td> <td>375 ms</td> </tr> <tr> <td>0000 0100</td> <td>4 ms</td> <td>500 ms</td> </tr> <tr> <td>...</td> <td>...</td> <td>...</td> </tr> <tr> <td>1111 1110</td> <td>254 ms</td> <td>31.750 sec</td> </tr> <tr> <td>1111 1111</td> <td>255 ms</td> <td>31.875 sec</td> </tr> </tbody> </table>	PULSE_GEN_1_ON_TIME	PULSE_GEN_1_ON_CLK		0	1	0000 0000	0 ms	0 ms	0000 0001	1 ms	125 ms	0000 0010	2 ms	250 ms	0000 0011	3 ms	375 ms	0000 0100	4 ms	500 ms	1111 1110	254 ms	31.750 sec	1111 1111	255 ms	31.875 sec
PULSE_GEN_1_ON_TIME	PULSE_GEN_1_ON_CLK																															
	0	1																														
0000 0000	0 ms	0 ms																														
0000 0001	1 ms	125 ms																														
0000 0010	2 ms	250 ms																														
0000 0011	3 ms	375 ms																														
0000 0100	4 ms	500 ms																														
...																														
1111 1110	254 ms	31.750 sec																														
1111 1111	255 ms	31.875 sec																														

PULSE_GEN_2_DELAY, 寄存器0x32**表59. PULSE_GEN_2_DELAY的位功能描述**

位	位的名称	访问类型	说明																													
[7:0]	PULSE_GEN_2_DELAY[7:0]	读/写	定义脉冲发生器2的第一个使能的第一个时钟的初始延迟时间。延迟表示为所选周期时钟速度的时钟周期数(参见寄存器0x35)。例如:																													
			<table border="1"> <thead> <tr> <th rowspan="2">PULSE_GEN_2_DELAY</th> <th colspan="2">PULSE_GEN_2_PRD_CLK</th> </tr> <tr> <th>0</th> <th>1</th> </tr> </thead> <tbody> <tr> <td>0000 0000</td> <td>0 ms</td> <td>0 ms</td> </tr> <tr> <td>0000 0001</td> <td>1 ms</td> <td>125 ms</td> </tr> <tr> <td>0000 0010</td> <td>2 ms</td> <td>250 ms</td> </tr> <tr> <td>0000 0011</td> <td>3 ms</td> <td>375 ms</td> </tr> <tr> <td>0000 0100</td> <td>4 ms</td> <td>500 ms</td> </tr> <tr> <td>...</td> <td>...</td> <td>...</td> </tr> <tr> <td>1111 1110</td> <td>254 ms</td> <td>31.750 sec</td> </tr> <tr> <td>1111 1111</td> <td>255 ms</td> <td>31.875 sec</td> </tr> </tbody> </table>	PULSE_GEN_2_DELAY	PULSE_GEN_2_PRD_CLK		0	1	0000 0000	0 ms	0 ms	0000 0001	1 ms	125 ms	0000 0010	2 ms	250 ms	0000 0011	3 ms	375 ms	0000 0100	4 ms	500 ms	1111 1110	254 ms	31.750 sec	1111 1111	255 ms	31.875 sec
PULSE_GEN_2_DELAY	PULSE_GEN_2_PRD_CLK																															
	0	1																														
0000 0000	0 ms	0 ms																														
0000 0001	1 ms	125 ms																														
0000 0010	2 ms	250 ms																														
0000 0011	3 ms	375 ms																														
0000 0100	4 ms	500 ms																														
...																														
1111 1110	254 ms	31.750 sec																														
1111 1111	255 ms	31.875 sec																														

ADP5586

PULSE_GEN_2_PERIOD, 寄存器0x33

表60. PULSE_GEN_2_PERIOD的位功能描述

位	位的名称	访问类型	说明																													
[7:0]	PULSE_GEN_2_PERIOD[7:0]	读/写	定义脉冲发生器2的周期。周期表示为所选周期时钟速度的时钟周期数 (参见寄存器0x35)。例如:																													
			<table border="1"> <thead> <tr> <th rowspan="2">PULSE_GEN_2_PERIOD</th> <th colspan="2">PULSE_GEN_2_PRD_CLK</th> </tr> <tr> <th>0</th> <th>1</th> </tr> </thead> <tbody> <tr> <td>0000 0000</td> <td>0 ms</td> <td>0 ms</td> </tr> <tr> <td>0000 0001</td> <td>1 ms</td> <td>125 ms</td> </tr> <tr> <td>0000 0010</td> <td>2 ms</td> <td>250 ms</td> </tr> <tr> <td>0000 0011</td> <td>3 ms</td> <td>375 ms</td> </tr> <tr> <td>0000 0100</td> <td>4 ms</td> <td>500 ms</td> </tr> <tr> <td>...</td> <td>...</td> <td>...</td> </tr> <tr> <td>1111 1110</td> <td>254 ms</td> <td>31.750 sec</td> </tr> <tr> <td>1111 1111</td> <td>255 ms</td> <td>31.875 sec</td> </tr> </tbody> </table>	PULSE_GEN_2_PERIOD	PULSE_GEN_2_PRD_CLK		0	1	0000 0000	0 ms	0 ms	0000 0001	1 ms	125 ms	0000 0010	2 ms	250 ms	0000 0011	3 ms	375 ms	0000 0100	4 ms	500 ms	1111 1110	254 ms	31.750 sec	1111 1111	255 ms	31.875 sec
PULSE_GEN_2_PERIOD	PULSE_GEN_2_PRD_CLK																															
	0	1																														
0000 0000	0 ms	0 ms																														
0000 0001	1 ms	125 ms																														
0000 0010	2 ms	250 ms																														
0000 0011	3 ms	375 ms																														
0000 0100	4 ms	500 ms																														
...																														
1111 1110	254 ms	31.750 sec																														
1111 1111	255 ms	31.875 sec																														

PULSE_GEN_2_ON_TIME, 寄存器0x34

表61. PULSE_GEN_2_ON_TIME的位功能描述

位	位的名称	访问类型	说明																													
[7:0]	PULSE_GEN_2_ON_TIME[7:0]	读/写	定义脉冲发生器2的开启时间。开启时间表示为所选时钟速度的时钟周期数 (参见寄存器0x35)。例如:																													
			<table border="1"> <thead> <tr> <th rowspan="2">PULSE_GEN_2_ON_TIME</th> <th colspan="2">PULSE_GEN_2_ON_CLK</th> </tr> <tr> <th>0</th> <th>1</th> </tr> </thead> <tbody> <tr> <td>0000 0000</td> <td>0 ms</td> <td>0 ms</td> </tr> <tr> <td>0000 0001</td> <td>1 ms</td> <td>125 ms</td> </tr> <tr> <td>0000 0010</td> <td>2 ms</td> <td>250 ms</td> </tr> <tr> <td>0000 0011</td> <td>3 ms</td> <td>375 ms</td> </tr> <tr> <td>0000 0100</td> <td>4 ms</td> <td>500 ms</td> </tr> <tr> <td>...</td> <td>...</td> <td>...</td> </tr> <tr> <td>1111 1110</td> <td>254 ms</td> <td>31.750 sec</td> </tr> <tr> <td>1111 1111</td> <td>255 ms</td> <td>31.875 sec</td> </tr> </tbody> </table>	PULSE_GEN_2_ON_TIME	PULSE_GEN_2_ON_CLK		0	1	0000 0000	0 ms	0 ms	0000 0001	1 ms	125 ms	0000 0010	2 ms	250 ms	0000 0011	3 ms	375 ms	0000 0100	4 ms	500 ms	1111 1110	254 ms	31.750 sec	1111 1111	255 ms	31.875 sec
PULSE_GEN_2_ON_TIME	PULSE_GEN_2_ON_CLK																															
	0	1																														
0000 0000	0 ms	0 ms																														
0000 0001	1 ms	125 ms																														
0000 0010	2 ms	250 ms																														
0000 0011	3 ms	375 ms																														
0000 0100	4 ms	500 ms																														
...																														
1111 1110	254 ms	31.750 sec																														
1111 1111	255 ms	31.875 sec																														

PULSE_GEN_CONFIG, 寄存器0x35**表62. PULSE_GEN_CONFIG的位功能描述**

位	位的名称	访问类型	说明
7	PULSE_GEN_1_INV	读/写	0 = 脉冲发生器1不反相。开启时间定义为高电平信号输出的时长。 1 = 脉冲发生器1输出反相。开启时间定义为高电平信号输出的时长。
6	PULSE_GEN_1_ON_CLK	读/写	定义脉冲发生器1开启时间所用的时钟速度。 0 = 1 ms。 1 = 125 ms。 不支持设置PULSE_GEN_1_ON_CLK = 1且PULSE_GEN_1_PRD_CLK = 0。
5	PULSE_GEN_1_PRD_CLK	读/写	定义脉冲发生器1周期所用的时钟速度。 0 = 1 ms。 1 = 125 ms。 不支持设置PULSE_GEN_1_ON_CLK = 1且PULSE_GEN_1_PRD_CLK = 0。
4	PULSE_GEN_1_EN	读/写	0 = 禁用脉冲发生器1。持续输出关闭信号。 1 = 使能脉冲发生器1。
3	PULSE_GEN_2_INV	读/写	0 = 脉冲发生器2不反相。开启时间定义为高电平信号输出的时长。 1 = 脉冲发生器2输出反相。开启时间定义为低电平信号输出的时长。
2	PULSE_GEN_2_ON_CLK	读/写	定义脉冲发生器2开启时间所用的时钟速度。 0 = 1 ms。 1 = 125 ms。 不支持设置PULSE_GEN_2_ON_CLK = 1且PULSE_GEN_2_PRD_CLK = 0。
1	PULSE_GEN_2_PRD_CLK	读/写	定义脉冲发生器2周期所用的时钟速度。 0 = 1 ms。 1 = 125 ms。 不支持设置PULSE_GEN_2_ON_CLK = 1且PULSE_GEN_2_PRD_CLK = 0。
0	PULSE_GEN_2_EN	读/写	0 = 禁用脉冲发生器2。持续输出关闭信号。 1 = 使能脉冲发生器2。

LOGIC_CFG, 寄存器0x36**表63. LOGIC_CFG的位功能描述**

位	位的名称	访问类型	说明
7	保留	保留	保留。
6	LY_INV	读/写	0 = LY输出在传入逻辑模块之前不反相。 1 = 逻辑模块的LY输出反相。
5	LC_INV	读/写	0 = LC输入在传入逻辑模块之前不反相。 1 = LC输入在传入逻辑模块之前反相。
4	LB_INV	读/写	0 = LB输入在传入逻辑模块之前不反相。 1 = LB输入在传入逻辑模块之前反相。
3	LA_INV	读/写	0 = LA输入在传入逻辑模块之前不反相。 1 = LA输入在传入逻辑模块之前反相。
[2:0]	LOGIC_SEL[2:0]	读/写	配置逻辑模块的数字复用器。参见图19。 000 = 关闭/禁用。 001 = AND。 010 = OR。 011 = XOR。 100 = FF。 101 = IN_LA。 110 = IN_LB。 111 = IN_LC。

ADP5586

LOGIC_FF_CFG, 寄存器0x37

表64. LOGIC_FF_CFG的位功能描述

位	位的名称	访问类型	说明
[7:2]	保留	保留	保留.
1	FF_SET	读/写	0 = 逻辑模块中的FF未置1。参见图19。 1 = 逻辑模块中的FF置1。
0	FF_CLR	读/写	0 = 逻辑模块中的FF未清0。参见图19。 1 = 逻辑模块中的FF清0。

LOGIC_INT_EVENT_EN, 寄存器0x38

表65. LOGIC_INT_EVENT_EN的位功能描述

位	位的名称	访问类型	说明
[7:3]	保留	保留	保留.
2	LY_DBNC_DIS	读/写	0 = 逻辑模块的输出在进入事件/中断模块之前去抖。 1 = 逻辑模块的输出在进入事件/中断模块之前未去抖。 谨慎使用, 因为毛刺可能导致过早产生中断。
1	LOGIC_EVENT_EN	读/写	0 = LY不能产生中断。 1 = 允许LY活动产生FIFO事件。
0	LOGIC_INT_LEVEL	读/写	配置产生中断的LY逻辑电平。 0 = LY低电平有效。 1 = LY高电平有效。

POLL_TIME_CFG, 寄存器0x39

表66. POLL_TIME_CFG的位功能描述

位	位的名称	访问类型	说明
[7:4]	保留	保留	保留.
3	PRECHARGE_TIME	读/写	定义预充电时间。 0 = 100 μ s。 1 = 200 μ s。
2	保留	保留	保留.
[1:0]	KEY_POLL_TIME[1:0]	读/写	配置连续扫描周期的间隔时间。 00 = 10 ms。 01 = 20 ms。 10 = 30 ms。 11 = 40 ms。

PIN_CONFIG_A, 寄存器0x3A

表67. PIN_CONFIG_A的位功能描述

位	位的名称	访问类型	说明
[7:6]	保留	保留	保留.
5	R5_CONFIG	读/写	0 = GPIO 6。 1 = 行5。
4	R4_CONFIG	读/写	0 = GPIO 5(其它配置参见表69中的R4_EXTEND_CFG, RESET)。 1 = 行4
3	R3_CONFIG	读/写	0 = GPIO 4(其它配置参见表69中的R3_EXTEND_CFG, LC)。 1 = 行3
2	R2_CONFIG	读/写	0 = GPIO 3 1 = 行2
1	R1_CONFIG	读/写	0 = GPIO 2 1 = 行1
0	R0_CONFIG	读/写	0 = GPIO 1/LY(其它配置参见表69中的R0_EXTEND_CFG, LY)。 1 = 行0

PIN_CONFIG_B, 寄存器0x3B**表68. PIN_CONFIG_B的位功能描述**

位	位的名称	访问类型	说明
[7:5]	保留	保留	保留。
4	C4_CONFIG	读/写	0 = GPIO 11。 1 = 列4。
3	C3_CONFIG	读/写	0 = GPIO 10。 1 = 列3。
2	C2_CONFIG	读/写	0 = GPIO 9。 1 = 列2。
1	C1_CONFIG_2	读/写	0 = GPIO 8(其它配置参见表69中的C1_EXTEND_CFG, PULSE_GEN_2)。 1 = 列1。
0	C0_CONFIG	读/写	0 = GPIO 7(其它配置参见表69中的C0_EXTEND_CFG, PULSE_GEN_1)。 1 = 列0。

PIN_CONFIG_C, 寄存器0x3C**表69. PIN_CONFIG_C的位功能描述**

位	位的名称	访问类型	说明
7	PULL_SELECT	读/写	0 = 键扫描期间行上拉使用300 kΩ电阻。 1 = 键扫描期间行上拉使用100 kΩ电阻。
6	C0_EXTEND_CFG	读/写	0 = C0保留配置为GPIO 7。 1 = C0重新配置为PULSE_GEN_1输出。
5	R4_EXTEND_CFG	读/写	0 = R4保留配置为GPIO 5。 1 = R4重新配置为RESET_OUT输出。
4	C1_EXTEND_CFG	读/写	0 = C1保留配置为GPIO 8。 1 = C1重新配置为PULSE_GEN_2输出。
3	R3_EXTEND_CFG	读/写	0 = R3保留配置为GPIO 4。 1 = R3重新配置为逻辑模块的LC输入。
[2:1]	保留	保留	保留。
0	R0_EXTEND_CFG	读/写	0 = R0保留配置为GPIO 1。 1 = R0重新配置为逻辑模块的LY输出。

GENERAL_CFG, 寄存器0x3D**表70. GENERAL_CFG的位功能描述**

位	位的名称	访问类型	说明
7	OSC_EN	读/写	0 = 禁用内部800 kHz振荡器。 1 = 使能内部800 kHz振荡器。
[6:5]	OSC_FREQ[1:0]	读/写	设置输入时钟频率由800 kHz基本振荡器提供给数字内核。频率越慢，则静态电流越低，但键和GPI扫描时间会增加。 00 = 50 kHz。 01 = 100 kHz。 10 = 200 kHz。 11 = 400 kHz。
[4:3]	保留	保留	保留。
2	SW_RESET	读/写	软件复位。置1复位ADP5586。此功能与拉低RST再拉高相似。器件重新编程前，至少等待200 μs。
1	INT_CFG	读/写	配置下述情况下INT引脚的行为：一个中断待处理时，用户试图清除该引脚。 0 = INT 引脚在中断待处理时保持置位。 1 = INT 引脚在中断待处理时解除置位50 μs，然后再重新置位。
0	RST_CFG	读/写	配置ADP5586对RST引脚和SW_RESET位的响应。 0 = RST为低电平时，ADP5586复位。 1 = RST为低电平时，ADP5586不复位。

ADP5586

INT_EN, 寄存器0x3E

表71. INT_EN的位功能描述

位	位的名称	访问类型	说明
[7:5]	保留	保留	保留.
4	LOGIC_IEN	读/写	0 = 禁用逻辑1中断。 1 = 如果LOGIC_INT位置1(寄存器0x01的位4), 则 $\overline{\text{INT}}$ 引脚置位。
3	保留	保留	保留.
2	OVRFLOW_IEN	读/写	0 = 禁用溢出中断。 1 = 如果OVRFLOW_INT位置1(寄存器0x01的位2), 则 $\overline{\text{INT}}$ 引脚置位。
1	GPI_IEN	读/写	0 = 禁用GPI中断。 1 = 如果GPI_INT位置1(寄存器0x01的位1), 则 $\overline{\text{INT}}$ 引脚置位。
0	EVENT_IEN	读/写	0 = 禁用事件中断。 1 = 如果EVENT_INT位置1(寄存器0x01的位0), 则 $\overline{\text{INT}}$ 引脚置位。

应用原理图

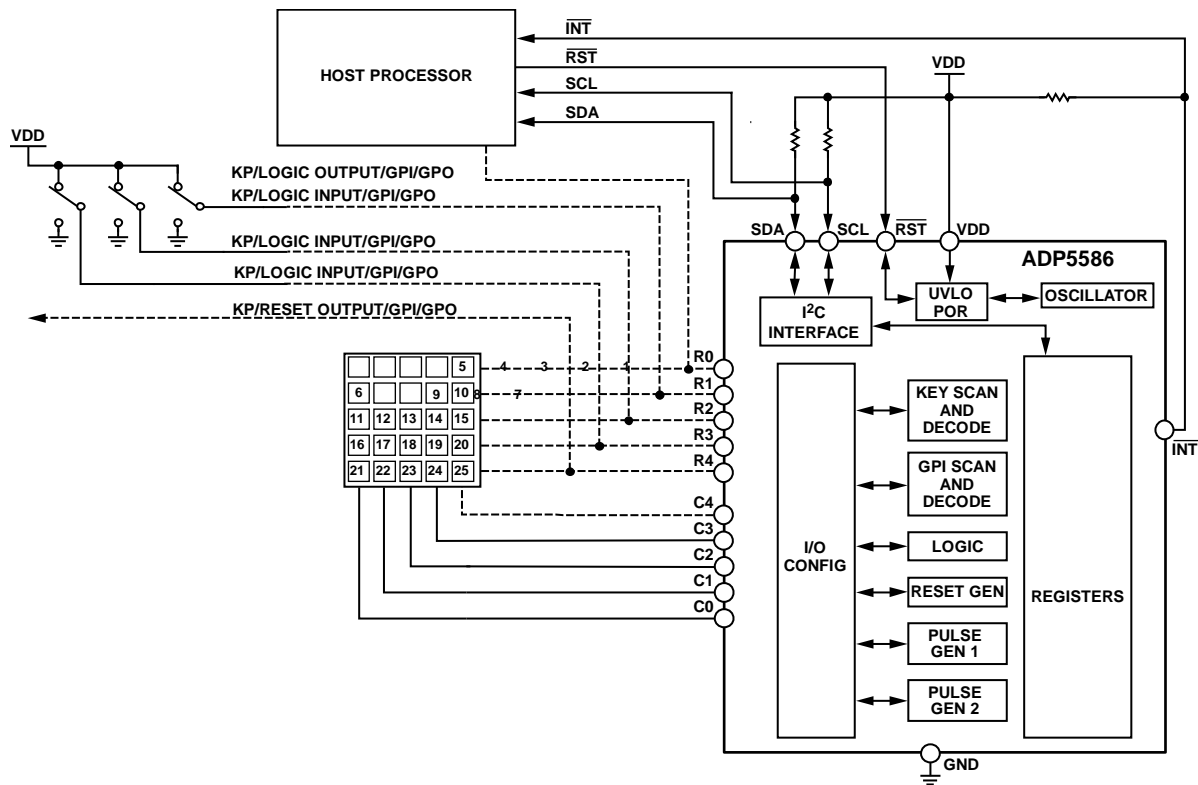
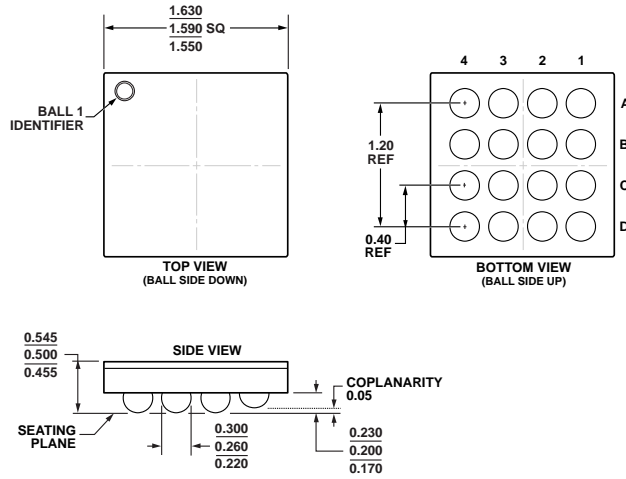


图28. 典型应用原理图

11/48-028

ADP5586

外形尺寸



01-20-2011-A

图29. 16引脚晶圆级芯片规模封装[WLCSP]
(CB-16-10)
尺寸单位: mm

订购指南

型号 ¹	温度范围	封装描述	封装选项
ADP5586ACBZ-00-R7	-40°C至+85°C	16引脚晶圆级芯片规模封装[WLCSP]	CB-16-10
ADP5586ACBZ-01-R7	-40°C至+85°C	16引脚晶圆级芯片规模封装[WLCSP]	CB-16-10
ADP5586ACBZ-03-R7	-40°C至+85°C	16引脚晶圆级芯片规模封装[WLCSP]	CB-16-10
ADP5586CB-EVALZ		WLCSP评估板	CB-16-10

¹Z = 符合RoHS标准的器件。

注释

注释

I²C指最初由Philips Semiconductors(现为NXP Semiconductors)开发的一种通信协议。