Chipsmall Limited consists of a professional team with an average of over 10 year of expertise in the distribution of electronic components. Based in Hongkong, we have already established firm and mutual-benefit business relationships with customers from,Europe,America and south Asia,supplying obsolete and hard-to-find components to meet their specific needs.

With the principle of “Quality Parts,Customers Priority,Honest Operation,and Considerate Service”,our business mainly focus on the distribution of electronic components. Line cards we deal with include Microchip,ALPS,ROHM,Xilinx,Pulse,ON,Everlight and Freescale. Main products comprise IC,Modules,Potentiometer,IC Socket,Relay,Connector.Our parts cover such applications as commercial,industrial, and automotives areas.

We are looking forward to setting up business relationship with you and hope to provide you with the best service and solution. Let us make a better world for our industry!

Contact us

Tel: +86-755-8981 8866 Fax: +86-755-8427 6832
Email & Skype: info@chipsmall.com Web: www.chipsmall.com
Address: A1208, Overseas Decoration Building, #122 Zhenhua RD., Futian, Shenzhen, China

Keypad Decoder and I/O Expansion

Data Sheet **ADP5585**

FEATURES

- **16-element FIFO for event recording**
- **10 configurable I/Os allowing functions such as**
  - **Key pad decoding for a matrix of up to 5 × 5**
    - **11 GPIOs (5 × 6) with ADP5585ACxZ-01-R7 models**
  - **Key press/release interrupts**
  - **GPIO functions**
  - **GPI with selectable interrupt level**
  - **100 kΩ or 300 kΩ pull-up resistors**
  - **300 kΩ pull-down resistors**
  - **GPO with push-pull or open-drain**
  - **Programmable logic block**
  - **PWM generator**
    - **Internal PWM generation**
    - **External PWM with internal PWM AND function**
  - **Reset generators**
- **I²C interface with fast mode plus (Fm+) support of up to 1 MHz**
- **Open-drain interrupt output**
- **16-ball WLCSP, 1.59 mm × 1.59 mm**
- **16-lead LFCSP, 3 mm × 3 mm**

FUNCTIONAL BLOCK DIAGRAM

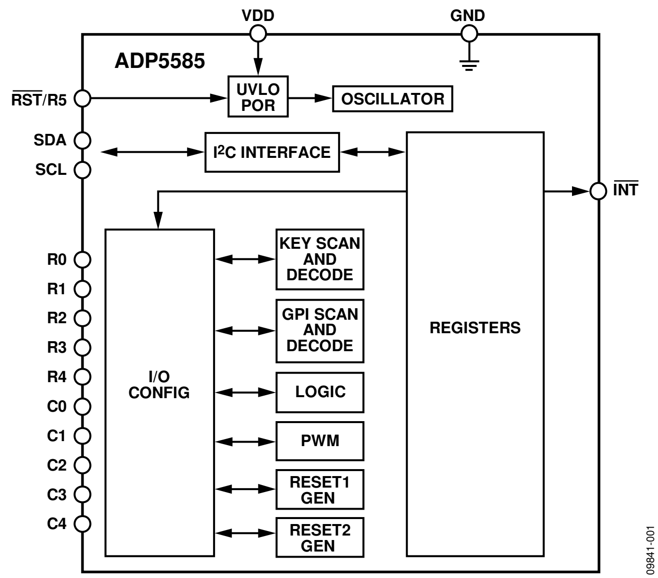


Figure 1.

APPLICATIONS

- **Keypad entries and input/output expansion capabilities**
- **Smart phones, remote controls, and cameras**
- **Healthcare, industrial, and instrumentation**

GENERAL DESCRIPTION

The ADP5585 is a 10 input/output port expander with a built in keypad matrix decoder, programmable logic, reset generator, and PWM generator. Input/output expander ICs are used in portable devices (phones, remote controls, and cameras) and nonportable applications (healthcare, industrial, and instrumentation). I/O expanders can be used to increase the number of I/Os available to a processor or to reduce the number of I/Os required through interface connectors for front panel designs.

The ADP5585 handles all key scanning and decoding and can flag the main processor via an interrupt line that new key events have occurred. GPI changes and logic changes can also be tracked as events via the FIFO, eliminating the need to monitor different registers for event changes. The ADP5585 is equipped with a FIFO to store up to 16 events. Events can be read back by the processor via an I²C-compatible interface.

The ADP5585 frees up the main processor from having to monitor the keypad, thereby reducing power consumption and/or increasing processor bandwidth for performing other functions.

The programmable logic functions allow common logic requirements to be integrated as part of the GPIO expander, thus saving board area and cost.

Rev. C Document Feedback

One Technology Way, P.O. Box 9106, Norwood, MA 02062-9106, U.S.A.
Tel: 781.329.4700

ADP5585* PRODUCT PAGE QUICK LINKS

Last Content Update: 02/23/2017

COMPARABLE PARTS

View a parametric search of comparable parts.

EVALUATION KITS

- ADP5585 Evaluation Board

DOCUMENTATION

Data Sheet

- ADP5585: Keypad Decoder and I/O Expansion Data Sheet

SOFTWARE AND SYSTEMS REQUIREMENTS

- ADP5589 Input Keyboard and GPIO Linux Driver

DESIGN RESOURCES

- ADP5585 Material Declaration
- PCN-PDN Information
- Quality And Reliability
- Symbols and Footprints

DISCUSSIONS

View all ADP5585 EngineerZone Discussions.

SAMPLE AND BUY

Visit the product page to see pricing options.

TECHNICAL SUPPORT

Submit a technical question or find your regional support number.

DOCUMENT FEEDBACK

Submit feedback for this data sheet.

TABLE OF CONTENTS

Features ... 1
Applications ... 1
Functional Block Diagram ... 1
General Description ... 1
Revision History ... 2
Specifications ... 3
Timing Diagram ... 4
Absolute Maximum Ratings ... 5
Thermal Resistance ... 5
ESD Caution ... 5
Pin Configurations and Function Descriptions ... 6
Theory of Operation ... 7
Device Enable ... 8
Device Overview ... 8
Functional Description ... 9
Event FIFO ... 9
Key Scan Control ... 9
GPI Input ... 12
GPO Output ... 12
Logic Blocks ... 12
PWM Block ... 13
Reset Blocks ... 14
Register Interface ... 15
Register Map ... 17
Detailed Register Descriptions ... 19
Applications Diagram ... 36
Outline Dimensions ... 37
Ordering Guide ... 38

REVISION HISTORY

1/13—Rev. B to Rev. C

Changes to Detailed Register Description Section ... 19
Changes to Table 31 and Table 32 ... 24
Changes to Table 33, Table 34, and Table 35 ... 25
Changes to Table 37 ... 26
Changes to Table 39 ... 27
Changes to Table 41 and Table 43 ... 28
Changes to Table 45 ... 29
Changes to Table 47 ... 30
Changes to Table 64 ... 34
Changes to Figure 27 ... 36

7/12—Rev. A to Rev B

Changes to Table 5 ... 8
Updated Outline Dimensions ... 36
Changes to Ordering Guide ... 37

10/11—Rev. Sp0 to Rev. A

Added 16-Lead LFCSP_WQ Package ... Universal
Changes to Features Section ... 1
Added Figure 4; Renumbered Sequentially ... 6
Changes to Table 4 ... 6
Changes to Device Enable Section and Table 5 ... 8
Change to General Section ... 11
Changes to Logic Blocks Section ... 12
Changes to PWM Block Section ... 13
Changes to Interrupts Section ... 14
Changes to Register Interface Section ... 15
Changes to Figure 27 ... 35
Updated Outline Dimensions ... 36
Changes to Ordering Guide ... 38

5/11—Revision Sp0: Initial Version

SPECIFICATIONS

VDD = 1.8 V to 3.3 V, $T_A = T_J = −40°C$ to +85°C, unless otherwise noted[1].

Table 1.

Parameter	Symbol	Test Conditions/Comments	Min	Typ	Max	Unit
SUPPLY VOLTAGE						
VDD Input Voltage Range	VDD		1.65		3.6	V
Undervoltage Lockout Threshold	$UVLO_{VDD}$	UVLO active, VDD falling	1.2	1.3		V
		UVLO inactive, VDD rising		1.4	1.6	V
SUPPLY CURRENT						
Standby Current	I_{STNBY}	VDD = 1.65 V		1	4	μA
		VDD = 3.3 V		1	10	μA
Operating Current (One Key Press)	I_{SCAN1}	Scan = 10 ms, CORE_FREQ = 50 kHz, scan active, 300 kΩ pull-up, VDD = 1.65 V		30	40	μA
	I_{SCAN2}	Scan = 10 ms, CORE_FREQ = 50 kHz, scan active, 100 kΩ pull-up, VDD = 1.65 V		35	45	μA
	I_{SCAN3}	Scan = 10 ms, CORE_FREQ = 50 kHz, scan active, 300 kΩ pull-up, VDD = 3.3 V		75	85	μA
	I_{SCAN4}	Scan = 10 ms, CORE_FREQ = 50 kHz, scan active, 100 kΩ pull-up, VDD = 3.3 V		80	90	μA
PULL-UP, PULL-DOWN RESISTANCE						
Pull-Up						
Option 1			50	100	150	kΩ
Option 2			150	300	450	kΩ
Pull-Down			150	300	450	kΩ
INPUT LOGIC LEVEL ($\overline{RST}$, SCL, SDA, R0, R1, R2, R3, R4, R5, C0, C1, C2, C3, C4)						
Input Voltage						
Logic Low	V_{IL}				0.3 VDD	V
Logic High	V_{IH}		0.7 VDD			V
Input Leakage Current (Per Pin)	$V_{I\text{-}Leak}$			0.1	1	μA
PUSH-PULL OUTPUT LOGIC LEVEL (R0, R1, R2, R3, R4, R5, C0, C1, C2, C3, C4)						
Output Voltage						
Logic Low	V_{OL1}	Sink current = 10 mA, maximum of five GPIOs active simultaneously			0.4	V
	V_{OL2}	Sink current = 10 mA, all GPIOs active simultaneously			0.5	V
Logic High	V_{OH}	Source current = 5 mA	0.7 VDD			V
Logic High Leakage Current (Per Pin)	$V_{OH\text{-}Leak}$			0.1	1	μA
OPEN-DRAIN OUTPUT LOGIC LEVEL ($\overline{INT}$, SDA)						
Output Voltage						
Logic Low						
$\overline{INT}$	V_{OL3}	I_{SINK} = 10 mA			0.4	V
SDA	V_{OL4}	I_{SINK} = 20 mA			0.4	V
Logic High Leakage Current (Per Pin)	$V_{OH\text{-}Leak}$			0.1	1	μA
Logic Propagation Delay				125	300	ns
FF Hold Time[2]				0		ns
FF Setup Time[2]				175		ns
GPIO Debounce[2]					70	μs
Internal Oscillator Frequency[3]	OSC_{FREQ}		900	1000	1100	kHz

Parameter	Symbol	Test Conditions/Comments	Min	Typ	Max	Unit
I²C TIMING SPECIFICATIONS						
Delay from UVLO/Reset Inactive to I²C Access					60	µs
SCL Clock Frequency	f_{SCL}		0		1000	kHz
SCL High Time	t_{HIGH}		0.26			µs
SCL Low Time	t_{LOW}		0.5			µs
Data Setup Time	$t_{SU; DAT}$		50			ns
Data Hold Time	$t_{HD; DAT}$		0			µs
Setup Time for Repeated Start	$t_{SU; STA}$		0.26			µs
Hold Time for Start/Repeated Start	$t_{HD; STA}$		0.26			µs
Bus Free Time for Stop and Start Condition	t_{BUF}		0.5			µs
Setup Time for Stop Condition	$t_{SU; STO}$		0.26			µs
Data Valid Time	$t_{VD; DAT}$				0.45	µs
Data Valid Acknowledge	$t_{VD; ACK}$				0.45	µs
Rise Time for SCL and SDA	t_R				120	ns
Fall Time for SCL and SDA	t_F				120	ns
Pulse Width of Suppressed Spike	t_{SP}		0		50	ns
Capacitive Load for Each Bus Line	C_B[4]				550	pF

[1] All limits at temperature extremes are guaranteed via correlation using standard statistical quality control (SQC). Typical values are at T_A = 25°C, VDD = 1.8 V.
[2] Guaranteed by design.
[3] All timers are referenced from the base oscillator and have the same ±10% accuracy.
[4] C_B is the total capacitance of one bus line in picofarads.

TIMING DIAGRAM

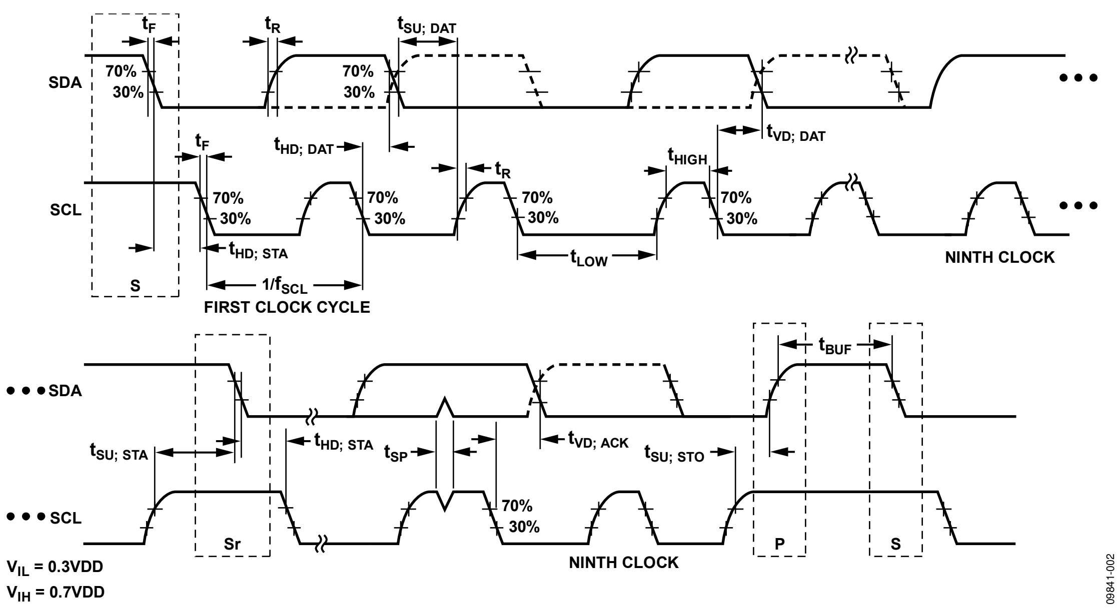


Figure 2. I²C Interface Timing Diagram

ABSOLUTE MAXIMUM RATINGS

Table 2.

Parameter	Rating
VDD to GND	−0.3 V to +4 V
SCL, SDA, $\overline{\text{RST}}$, $\overline{\text{INT}}$, R0, R1, R2, R3, R4, C0, C1, C2, C3, C4 to GND	−0.3 V to (VDD + 0.3 V)
Temperature Range	
Operating (Ambient)	−40°C to +85°C[1]
Operating (Junction)	−40°C to +125°C
Storage	−65°C to +150°C

[1] In applications where high power dissipation and poor thermal resistance are present, the maximum ambient temperature may need to be derated. Maximum ambient temperature ($T_{A\,(MAX)}$) is dependent on the maximum operating junction temperature ($T_{J\,(MAXOP)}$ = 125°C), the maximum power dissipation of the device ($P_{D\,(MAX)}$), and the junction-to-ambient thermal resistance of the device/package in the application (θ_{JA}), using the following equation: $T_{A\,(MAX)} = T_{J\,(MAXOP)} - (\theta_{JA} \times P_{D\,(MAX)})$.

Stresses above those listed under Absolute Maximum Ratings may cause permanent damage to the device. This is a stress rating only; functional operation of the device at these or any other conditions above those indicated in the operational section of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

Absolute maximum ratings apply individually only, not in combination. Unless otherwise specified, all other voltages are referenced to GND.

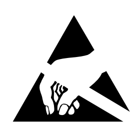
THERMAL RESISTANCE

θ_{JA} is specified for the worst-case conditions, that is, a device soldered in a printed circuit board (PCB) for surface-mount packages.

Table 3.

Thermal Resistance	θ_{JA}	Unit
16-Ball WLCSP	62	°C/W
Maximum Power Dissipation	70	mW
16-Lead LFCSP	67.154	°C/W
Maximum Power Dissipation	70	mW

ESD CAUTION



ESD (electrostatic discharge) sensitive device. Charged devices and circuit boards can discharge without detection. Although this product features patented or proprietary protection circuitry, damage may occur on devices subjected to high energy ESD. Therefore, proper ESD precautions should be taken to avoid performance degradation or loss of functionality.

PIN CONFIGURATIONS AND FUNCTION DESCRIPTIONS

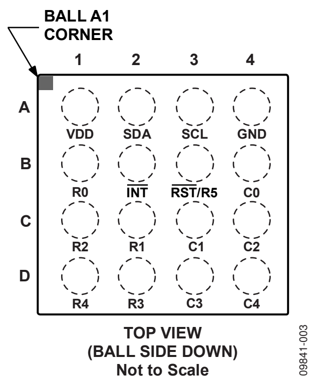


Figure 3. WLCSP Pin Configuration

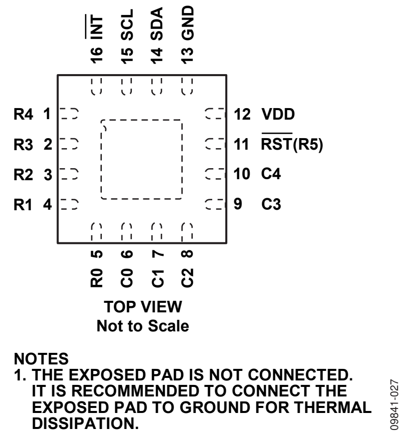


Figure 4. LFCSP Pin Configuration

Table 4. Pin Function Descriptions

Pin No.			
WLCSP	**LFCSP**	**Mnemonic**	**Description**
D1	1	R4	GPIO 5 (GPIO Alternate Function: RESET1). This pin functions as Row 4 when used as a keypad.
D2	2	R3	GPIO 4 (GPIO Alternate Function: Logic Block Input LC, PWM_OUT). This pin functions as Row 3 when used as a keypad.
C1	3	R2	GPIO 3 (GPIO Alternate Function: Logic Block Input LB). This pin functions as Row 2 when used as a keypad.
C2	4	R1	GPIO 2 (GPIO Alternate Function: Logic Block Input LA). This pin functions as Row 1 when used as a keypad.
B1	5	R0	GPIO 1 (GPIO Alternate Function: Logic Block Output LY). This pin functions as Row 0 when used as a keypad.
B4	6	C0	GPIO 7. This pin functions as Column 0 when used as a keypad.
C3	7	C1	GPIO 8. This pin functions as Column 1 when used as a keypad.
C4	8	C2	GPIO 9. This pin functions as Column 2 when used as a keypad.
D3	9	C3	GPIO 10 (GPIO Alternate Function: PWM_IN). This pin functions as Column 3 when used as a keypad.
D4	10	C4	GPIO 11 (GPIO Alternate Function: RESET2). This pin functions as Column 4 when used as a keypad.
B3	11	$\overline{\text{RST}}$/R5	Input Reset Signal. To expand the keypad matrix, select the ADP5585ACBZ-01-R7 or the ADP5585ACPZ-01-R7 device model for this pin to function as GPIO 6/Row 5.
A1	12	VDD	Supply Voltage Input.
A4	13	GND	Ground.
A2	14	SDA	I^2C Data Input/Output.
A3	15	SCL	I^2C Clock Input.
B2	16	$\overline{\text{INT}}$	Open-Drain Interrupt Output.
	EP	EP	Exposed Pad. The exposed pad is not connected. It is recommended to connect the exposed pad to ground for thermal dissipation.

THEORY OF OPERATION

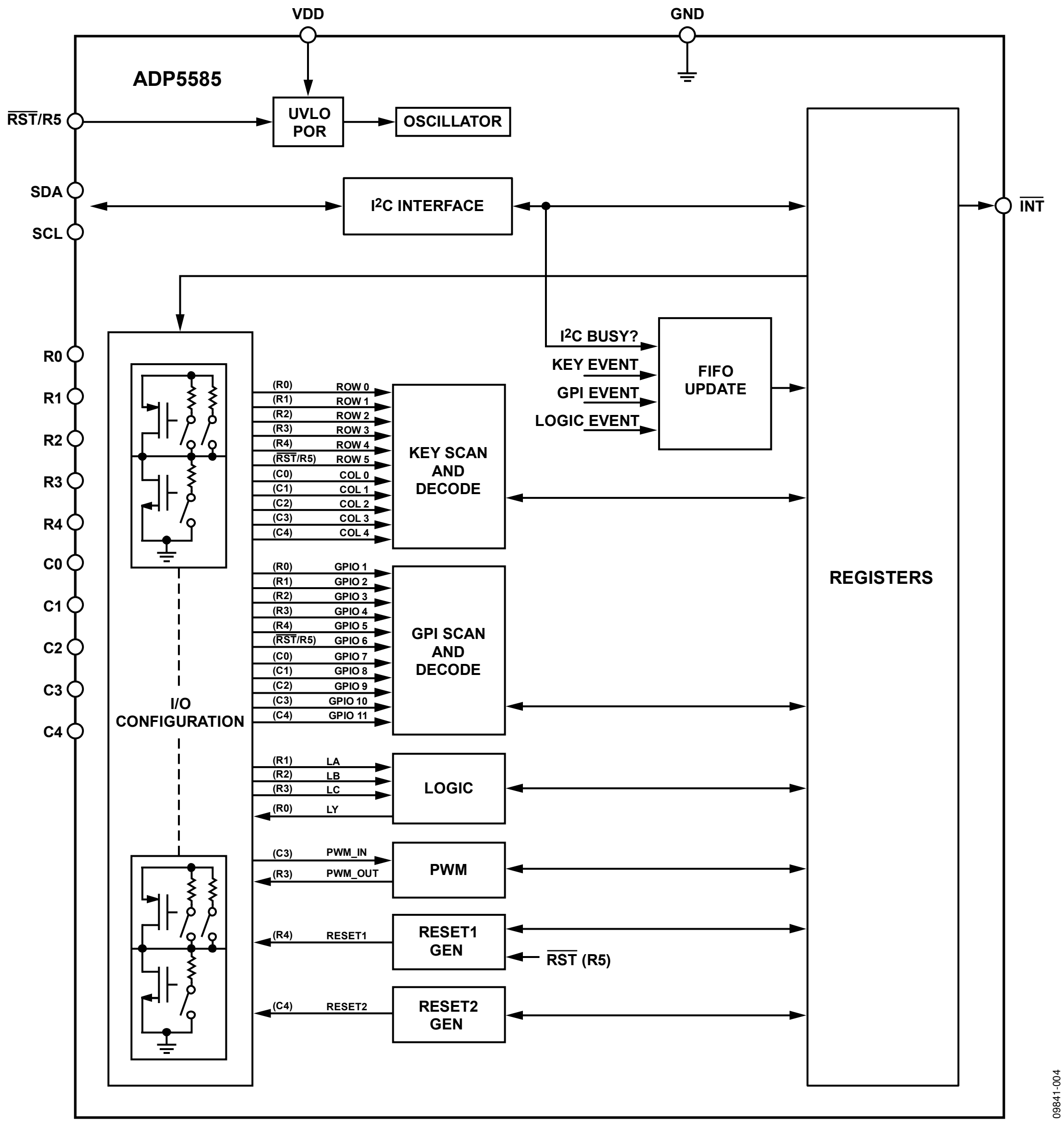


Figure 5. Internal Block Diagram

DEVICE ENABLE

When sufficient voltage is applied to VDD and the $\overline{\text{RST}}$ pin is driven with a logic high level, the ADP5585 starts up in standby mode with all settings at default. The user can configure the device via the I^{2}C interface. When the $\overline{\text{RST}}$ pin is low, the ADP5585 enters a reset state and all settings return to default. The $\overline{\text{RST}}$ pin features a debounce filter.

If using the ADP5585ACBZ-01-R7 or ADP5585ACPZ-01-R7 device model, the $\overline{\text{RST}}$ pin acts as an extra row pin. Without a reset pin, the only method to reset the device is by bringing VDD below the UVLO threshold.

DEVICE OVERVIEW

The ADP5585 contains 10 multiconfigurable input/output pins. Each pin can be programmed to enable the device to carry out its various functions, as follows:

- Keypad matrix decoding (five-column by five-row matrix maximum).
- General-purpose I/O expansion (up to 10 inputs/outputs).
- PWM generation.
- Logic function building blocks (up to three inputs and one output).
- Two reset generators.

All 10 input/output pins have an I/O structure as shown in Figure 6.

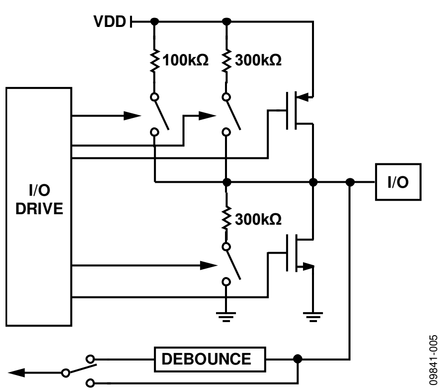


Figure 6. I/O Structure

Each I/O can be pulled up with a 100 kΩ or 300 kΩ resistor or pulled down with a 300 kΩ resistor. For logic output drive, each I/O has a 5 mA PMOS source and a 10 mA NMOS sink for a push-pull type output. For open-drain output situations, the 5 mA PMOS source is not enabled. For logic input applications, each I/O can be sampled directly or, alternatively, sampled through a debounce filter.

The I/O structure shown in Figure 6 allows for all GPI and GPO functions, as well as PWM and clock divide functions. For key matrix scan and decode, the scanning circuit uses the 100 kΩ or 300 kΩ resistor for pulling up keypad row pins and the 10 mA NMOS sinks for grounding keypad column pins (see the Key Scan Control section for details about key decoding).

Configuration of the device is carried out by programming an array of internal registers via the I^{2}C interface. Feedback of device status and pending interrupts can be flagged to an external processor by using the $\overline{\text{INT}}$ pin.

The ADP5585 is offered with three feature sets. Table 5 lists the options that are available for each model of the ADP5585.

Table 5. Matrix Options by Device Model

Model	Description
ADP5585ACBZ-00-R7	GPIO pull up (default option) 5-row × 5-column matrix
ADP5585ACBZ-01-R7	Row 5 added to GPIOs 6-row × 5-column matrix
ADP5585ACBZ-02-R7	No pull-up resistors to special function pins[1] 5-row × 5-column matrix
ADP5585ACBZ-04-R7	Pull-down resistors to all GPIO pins on start-up 5-row × 5-column matrix
ADP5585ACPZ-00-R7	GPIO pull up (default option) 5-row × 5-column matrix
ADP5585ACPZ-01-R7	Row 5 added to GPIOs 6-row × 5-column matrix
ADP5585ACPZ-03-R7	Alternate I^{2}C address (0x30) 5-row × 5-column matrix

[1] Special function pins are defined as R0, R3, R4, and C4. See Table 4 for details.

FUNCTIONAL DESCRIPTION

EVENT FIFO

Before going into detail on the various ADP5585 blocks, it is important to understand the function of the event FIFO. The ADP5585 features an event FIFO that can record as many as 16 events. By default, the FIFO primarily records key events, such as key press and key release. However, it is possible to configure the general-purpose input (GPI) and logic activity to generate event information on the FIFO as well. An event count, EC[4:0], is composed of five bits and works in tandem with the FIFO so that the user knows how much of the FIFO must be read back at any given time.

The FIFO is composed of 16 eight-bit sections that the user accesses by reading the FIFO_x registers. The actual FIFO is not in user accessible registers until a read occurs. The FIFO can be thought of as a "first in first out" buffer that is used to fill Register 0x03 to Register 0x12.

The event FIFO is made up of 16 eight-bit registers. In each register, Bits[6:0] hold the event identifier, and Bit 7 holds the event state. With seven bits, 127 different events can be identified. See Table 11 for event decoding.

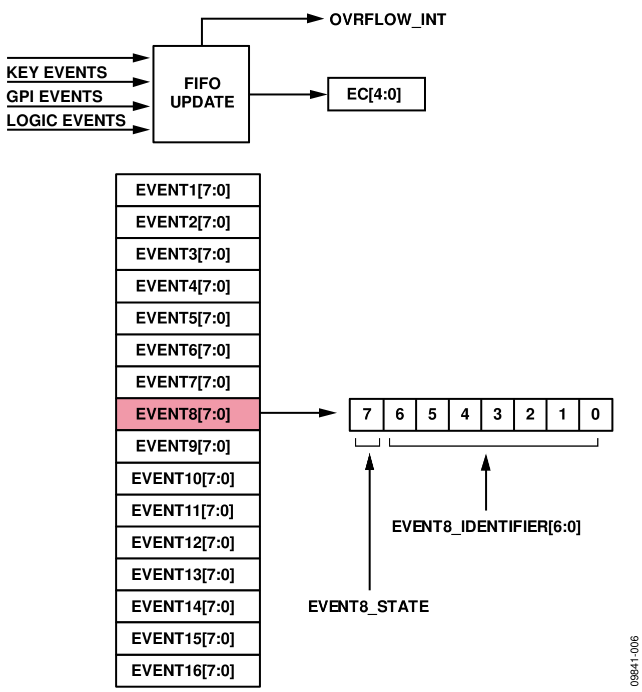


Figure 7. Breakdown of Eventx[7:0] Bits

When events are available on the FIFO, the user should first read back the event count, EC[4:0], to determine how many events must be read back. Events can be read from the top of the FIFO only. When an event is read back, all remaining events in the FIFO are shifted up one location, and the EC[4:0] count is decremented.

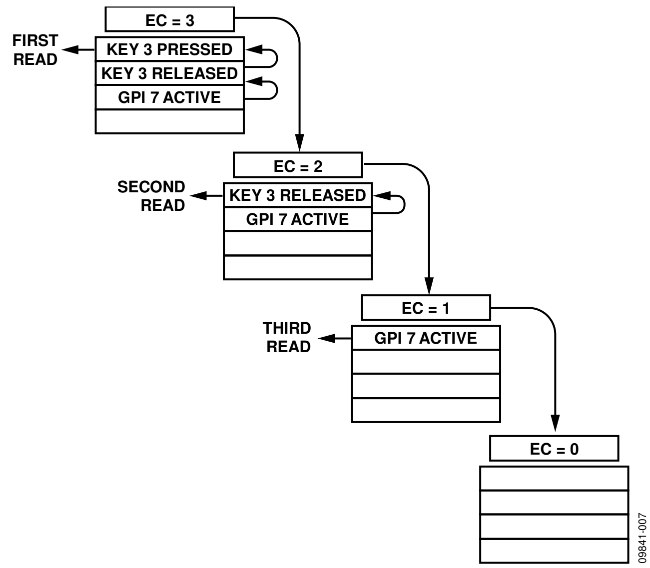


Figure 8. FIFO Operation

The FIFO registers (0x03 to 0x12) always point to the top of the FIFO (that is, the location of EVENT1[7:0]). If the user tries to read back from any location in a FIFO, data is always obtained from the top of that FIFO. This ensures that events can only be read back in the order in which they occurred, thus ensuring the integrity of the FIFO system.

As stated above, some of the onboard functions of ADP5585 can be programmed to generate events on the FIFO. A FIFO update control block manages updates to the FIFO. If an I^2C transaction is accessing any of the FIFO address locations, updates are paused until the I^2C transaction has completed.

A FIFO overflow event occurs when more than 16 events are generated prior to an external processor reading a FIFO and clearing it.

If an overflow condition occurs, the overflow status bit is set. An interrupt is generated if overflow interrupt is enabled, signaling to the processor that more than 16 events have occurred.

KEY SCAN CONTROL

General

The 10 input/output pins can be configured to decode a keypad matrix up to a maximum size of 25 switches (5 × 5 matrix). Smaller matrices can also be configured, freeing up the unused row and column pins for other I/O functions.

The R0 through R4 I/O pins comprise the rows of the keypad matrix. The C0 through C4 I/O pins comprise the columns of the keypad matrix. Pins used as rows are pulled up via the internal 300 kΩ (or 100 kΩ) resistors. Pins used as columns are driven low via the internal NMOS current sink.

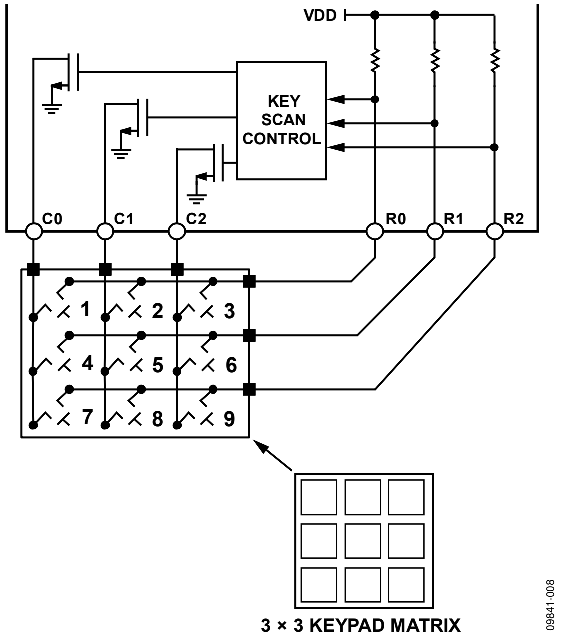


Figure 9. Simplified Key Scan Block

Figure 9 shows a simplified representation of the key scan block using three row and three column pins connected to a small 3 × 3, nine-switch keypad matrix. When the key scanner is idle, the row pins are pulled high and the column pins are driven low. The key scanner operates by checking the row pins to see if they are low.

If Switch 6 in the matrix is pressed, R1 connects to C2. The key scan circuit senses that one of the row pins has been pulled low, and a key scan cycle begins. Key scanning involves driving all column pins high, then driving each column pin, one at a time, low and sensing whether a row pin is low or not. All row/column pairs are scanned; therefore, if multiple keys are pressed, they are detected.

To prevent glitches or narrow press times being registered as a valid key press, the key scanner requires the key be pressed for two scan cycles. The key scanner has a wait time between each scan cycle; therefore, the key must be pressed and held for at least this wait time to register as being pressed. If the key is continuously pressed, the key scanner continues to scan, wait, scan, wait, and so forth.

If Switch 6 is released, the connection between R1 and C2 breaks, and R1 is pulled up high. The key scanner requires that the key be released for two scan cycles because the release of a key is not necessarily in sync with the key scanner, it may take up to two full wait/scan cycles for a key to register as released. When the key is registered as released, and no other keys are pressed, the key scanner returns to idle mode.

For the remainder of this document, the press/release status of a key is represented as simply a logic signal in the figures. A logic high level represents the key status as pressed, and a logic low represents released. This eliminates the need to draw individual row/column signals when describing key events.

Figure 10. Logic Low: Released, Logic High: Pressed

Figure 11 shows a detailed representation of the key scan block and its associated control and status signals. When all row and column pins are used, a matrix of 25 unique keys can be scanned.

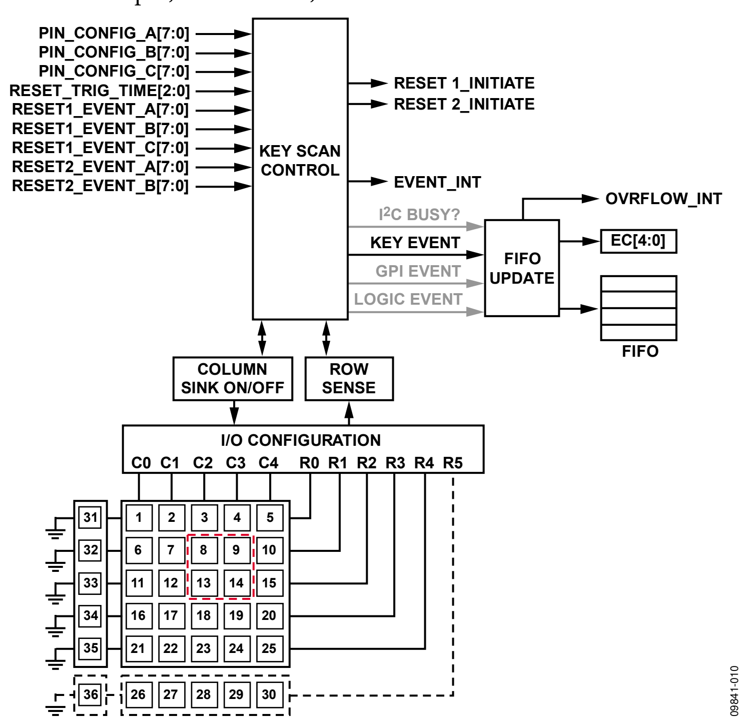


Figure 11. Detailed Key Scan Block

Use Registers PIN_CONFIG_A[7:0] and PIN_CONFIG_B[7:0] to configure I/Os for keypad decoding. The number label on each key switch represents the event identifier that is recorded if that switch was pressed. If all row/column pins are configured, it is possible to observe all 25 key identifiers on the FIFO. A larger 6 × 5 matrix can be configured by using the ADP5585ACBZ-01-R7 or the ADP5585ACPZ-01-R7.

If a smaller 2 × 2 matrix is configured, for example, by using the C2 and C3 column pins and the R1 and R2 row pins, only the four event identifiers (8, 9, 13, and 14) can possibly be observed on the FIFO, as shown in Figure 11.

By default, ADP5585 records key presses and releases on the FIFO. Figure 12 illustrates what happens when a single key is pressed and released. Initially, the key scanner is idle. When Key 3 is pressed, the scanner begins scanning through all configured row/column pairs. After the scan wait time, the scanner again scans through all configured row/column pairs and detects that Key 3 has remained pressed, which sets the EVENT_INT interrupt. The event counter, EC[4:0], is incremented to 1, EVENT1_IDENTIFIER[6:0] of the FIFO is updated with its event identifier set to 3, and its EVENT1_STATE bit is set to 1, indicating a press.

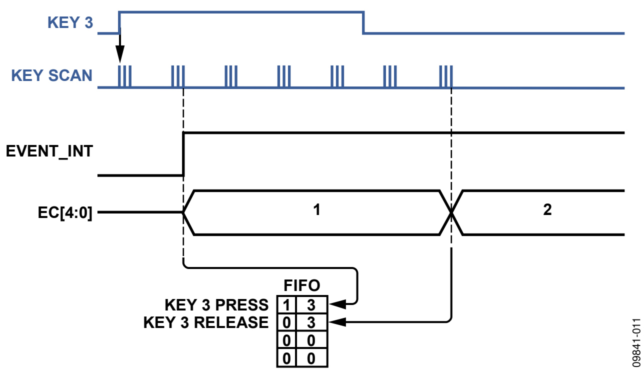


Figure 12. Press and Release Event

The key scanner continues the scan/wait cycles while the key remains pressed. If the scanner detects that the key has been released for two consecutive scan cycles, the event counter, EC[4:0], is incremented to 2, and EVENT2_IDENTIFIER[6:0] of the FIFO is updated with its event identifier set to 3. Its EVENT2_STATE bit is set to 0, indicating a release. The key scanner returns to idle mode because no other keys are pressed.

The EVENT_INT interrupt can be triggered by both press and release key events. As shown in Figure 14, if Key 3 is pressed, EVENT_INT is asserted, EC[4:0] is updated, and the FIFO is updated. During the time that the key remains pressed, it is possible for the FIFO to be read, the event counter decremented to 0, and EVENT_INT cleared. When the key is finally released, EVENT_INT is asserted, the event counter is incremented, and the FIFO is updated with the release event information.

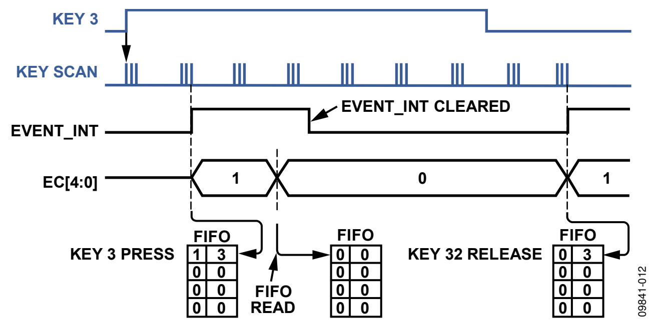


Figure 13. Asserting the EVENT_INT Interrupt Key Pad Extension

As shown in Figure 11, the keypad can be extended if each row is connected directly to ground by a switch. If the switch placed between R0 and ground is pressed, the entire row is grounded. When the key scanner completes scanning, it normally detects Key 1 to Key 5 as being pressed; however, this unique condition is decoded by the ADP5585, and Key Event 31 is assigned to it. Up to eight more key event assignments are possible, allowing the keypad size to extend up to 30. However, if one of the extended keys is pressed, none of the keys on that row is detectable. Activation of a ground key causes all other keys sharing that row to be undetectable.

Ghosting

Ghosting is an occurrence where, given certain key press combinations on a keypad matrix, a false positive reading of an additional key is detected. Ghosting is created when three or more keys are pressed simultaneously on multiple rows or columns (see Figure 14). Key combinations that form a right angle on the keypad matrix can cause ghosting.

The solution to ghosting is to select a keypad matrix layout that takes into account three key combinations that are most likely to be pressed together. Multiple keys pressed across one row or across one column do not cause ghosting. Staggering keys so that they do not share a column also avoids ghosting. The most common practice is to place keys that are likely to be pressed together in the same row or column. Some examples of keys that are likely to be pressed together are as follows:

- The navigation keys in combination with Select.
- The navigation keys in combination with the space bar.
- The reset combination keys, such as CTRL + ALT + DEL.

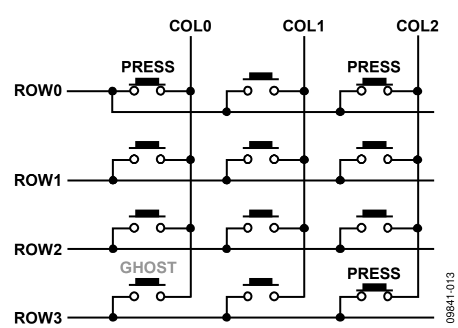


Figure 14. COL0: ROW3 is a Ghost Key Due to a Short Among ROW0, COL0, COL2, and ROW3 During Key Press

GPI INPUT

Each of the 10 input/output lines can be configured as a general-purpose logic input line. Figure 15 shows a detailed representation of the GPI scan and detect block and its associated control and status signals.

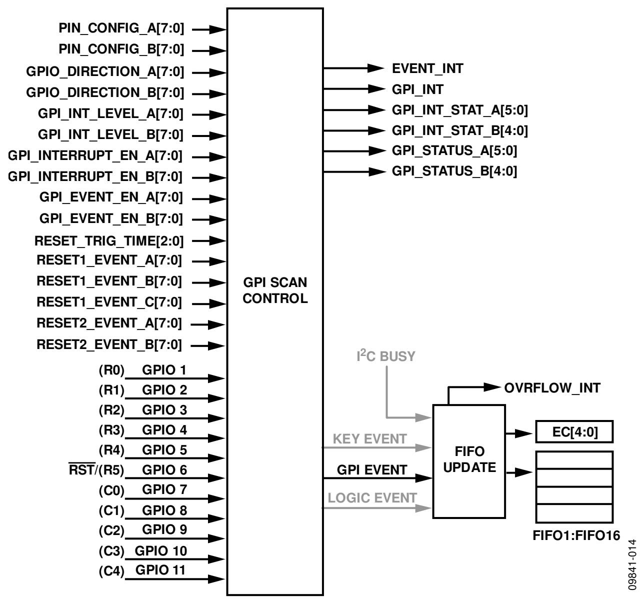


Figure 15. GPI Scan and Detect Block

The current input state of each GPI can be read back using the GPI_STATUS_x registers. Each GPI can be programmed to generate an interrupt via the GPI_INTERRUPT_EN_x registers. The interrupt status is stored in the GPI_INT_STAT_x registers. GPI interrupts can be programmed to trigger on the positive or negative edge by configuring the GPI_INT_LEVEL_x registers. If any of the GPI interrupts is triggered, the master GPI_INT interrupt is also triggered. Figure 16 shows a single GPI and how it affects its corresponding status and interrupt status bits.

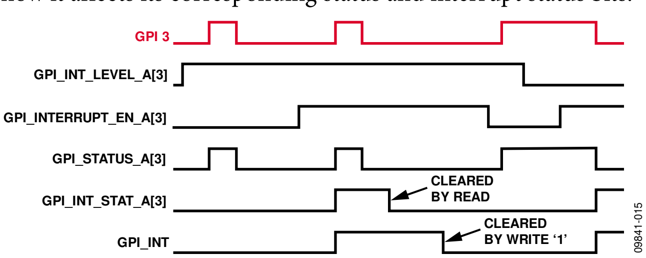


Figure 16. Single GPI Example

GPIs can be programmed to generate FIFO events via the GPI_EVENT_EN_x registers. GPIs in this mode do not generate GPI_INT interrupts and instead generate EVENT_INT interrupts. Figure 17 shows several GPI lines and their effects on the FIFO and event count, EC[4:0].

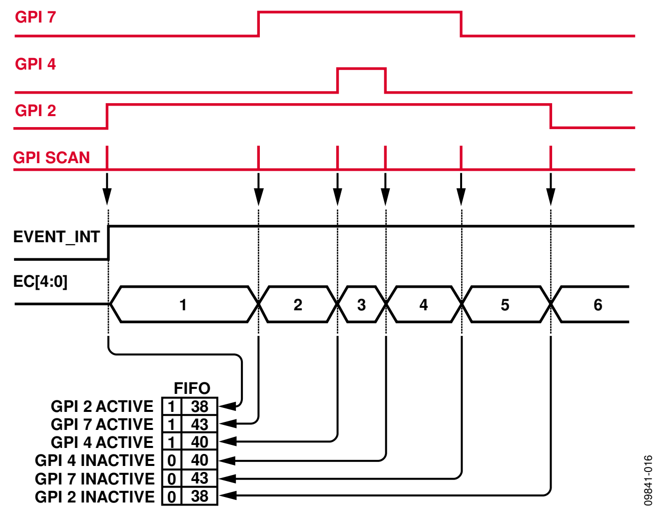


Figure 17. Multiple GPI Example

The GPI scanner is idle until it detects a level transition. It scans the GPI inputs and updates accordingly. It then returns to idle immediately, it does not scan/wait, like the key scanner. As such, the GPI scanner can detect narrow pulses once they get past the 50 µs input debounce filter.

GPO OUTPUT

Each of the 10 input/output lines can be configured as a general-purpose output (GPO) line. Figure 6 shows a detailed diagram of the I/O structure. See the Detailed Register Descriptions section for GPO configuration and usage.

LOGIC BLOCKS

Several of the ADP5585 input/output lines can be used as inputs and outputs for implementing some common logic functions.

The R1, R2, and R3 input/output pins can be used as inputs, and the R0 input/output pin can be used as an output for the logic block.

The outputs from the logic blocks can be configured to generate interrupts. They can also be configured to generate events on the FIFO.

Figure 19 shows a detailed diagram of the internal make-up of the logic block, illustrating the possible logic functions that can be implemented.

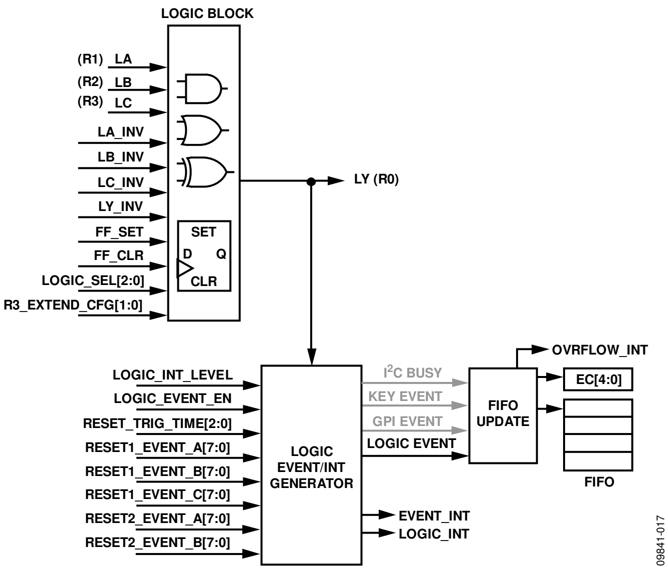


Figure 18. Logic Block Overview

PWM BLOCK

The ADP5585 features a PWM generator whose output can be configured to drive out on the R3 I/O pin. PWM on/off times are programmed via four 8-bit registers (see Figure 20). Each bit of the on or off time represents 1 µs. The highest frequency obtainable from the PWM is performed by setting the least significant bit of both the on and off time bit patterns, resulting in a 500 kHz signal with a 50% duty cycle.

The PWM block provides support for continuous PWM mode as well as a one-shot mode (see Table 59). Additionally, an external signal can be AND'ed with the internal PWM signal. This option can be selected by writing a 1 to PWM_IN_AND (PWM_CFG[2]). The input to the external AND is the C3 I/O pin. C3 should be set to GPI. Note that the debounce for C3 results in a delay of the AND'ing, and can be turned on or off using Register 0x21.

Newly programmed values are not latched until the final byte, PWM_ONT_HIGH_BYTE (Register 0x32, Bits[7:0]), is written.

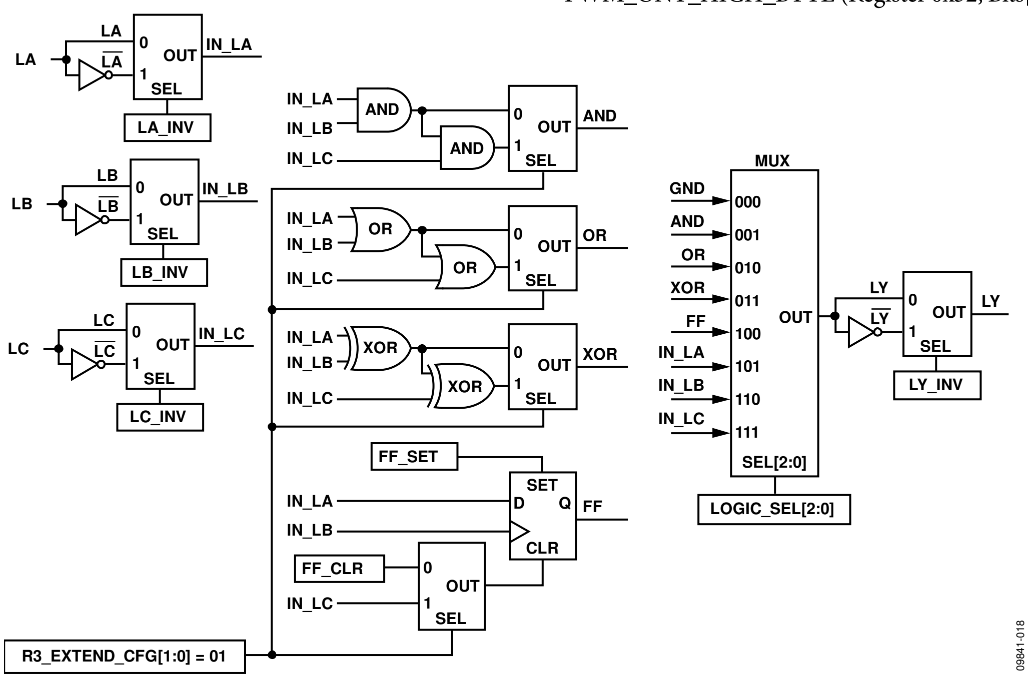


Figure 19. Logic Block

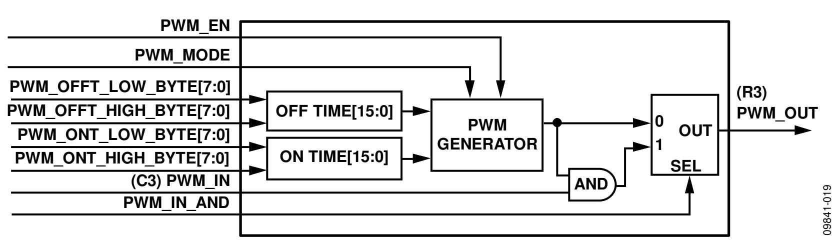


Figure 20. PWM Block Diagram

RESET BLOCKS

ADP5585 features two reset blocks that can generate reset conditions if certain events are detected simultaneously. Up to three reset trigger events can be programmed for RESET1. Up to two reset trigger events can be programmed for RESET2. The event scan control blocks monitor whether these events are present for the duration of RESET_TRIG_TIME[2:0] (Register 0x2E, Bits[4:2]). If they are, reset-initiate signals are sent to the reset generator blocks. The generated reset signal pulse width is programmable.

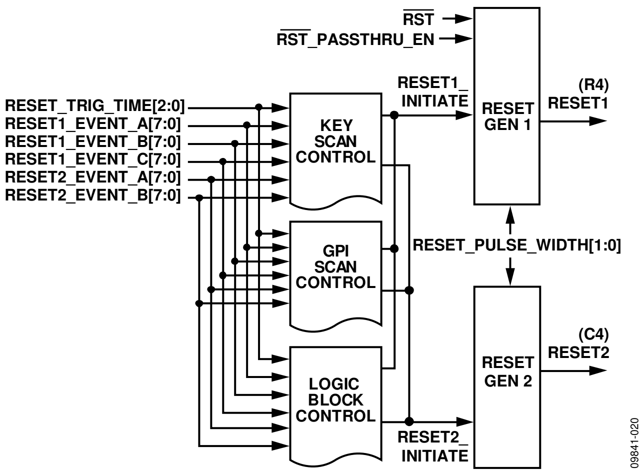


Figure 21. Reset Blocks

The Reset 1 signal uses the R4 I/O pin as its output. A pass through mode allows the main $\overline{\text{RST}}$ pin to be output on the R4 pin also. The Reset 2 signal uses the C4 I/O pin as its output.

The reset generation signals are useful in situations where the system processor has locked up and the system is unresponsive to input events. The user can press one of the reset event combinations and initiate a system wide reset. This alleviates the need for removing the battery from the system and doing a hard reset.

It is not recommended to use the immediate trigger time (see Table 54) because this setting may cause false triggering.

Interrupts

The $\overline{\text{INT}}$ pin can be asserted low if any of the internal interrupt sources is active. The user can select which internal interrupts interact with the external interrupt pin in Register 0x3C (refer to Table 68). Register 0x3B allows the user to choose whether the external interrupt pin remains asserted, or deasserts for 50 μs, then reasserts, in the case that there are multiple internal interrupts asserted and one is cleared (refer to Table 67).

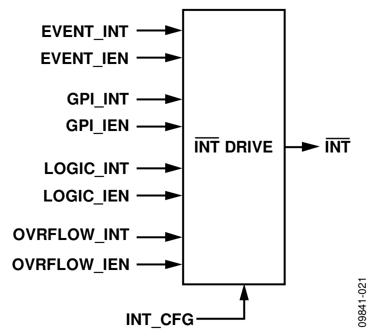


Figure 22. Asserting $\overline{\text{INT}}$ Low

REGISTER INTERFACE

Register access to the ADP5585 is acquired via its I^{2}C-compatible serial interface. The interface can support clock frequencies of up to 1 MHz. If the user is accessing the FIFO or key event counter (KEC), FIFO/KEC updates are paused. If the clock frequency is very low, events may not be recorded in a timely manner. FIFO or KEC updates can happen up to 23 µs after an interrupt is asserted because of the number of I^{2}C cycles required to perform an I^{2}C read or write. This delay should not present an issue to the user.

Figure 23 shows a typical write sequence for programming an internal register. The cycle begins with a start condition, followed by the hard coded 7-bit device address, which for the ADP5585 is 0x34, followed by the R/$\overline{W}$ bit set to 0 for a write cycle. The ADP5585 acknowledges the address byte by pulling the data line low. The address of the register to which data is to be written is sent next. The ADP5585 acknowledges the register pointer byte by pulling the data line low. The data byte to be written is sent next. The ADP5585 acknowledges the data byte by pulling the data line low. A stop condition completes the sequence.

Figure 24 shows a typical multibyte write sequence for programming internal registers. The cycle begins with a start condition followed by the 7-bit device address (0x34 for all models except the ADP5585ACPZ-03-R7, 0x30 for the ADP5585ACPZ-03-R7 only), followed by the R/$\overline{W}$ bit set to 0 for a write cycle. The ADP5585 acknowledges the address byte by pulling the data line low. The address of the register to which data is to be written is sent next. The ADP5585 acknowledges the register pointer byte by pulling the data line low. The data byte to be written is sent next. The ADP5585 acknowledges the data byte by pulling the data line low. The pointer address is then incremented to write the next data byte, until it finishes writing the n data byte. The ADP5585 pulls the data line low after every byte, and a stop condition completes the sequence.

Figure 25 shows a typical byte read sequence for reading internal registers. The cycle begins with a start condition followed by the 7-bit device address (0x34 for all models except the ADP5585ACPZ-03-R7, 0x30 for the ADP5585ACPZ-03-R7 only), followed by the R/$\overline{W}$ bit set to 0 for a write cycle. The ADP5585 acknowledges the address byte by pulling the data line low. The address of the register from which data is to be read is sent next. The ADP5585 acknowledges the register pointer byte by pulling the data line low. A start condition is repeated, followed by the 7-bit device address (0x34 for all models except the ADP5585ACPZ-03-R7, 0x30 for the ADP5585ACPZ-03-R7 only), followed by the R/$\overline{W}$ bit set to 1 for a read cycle. The ADP5585 acknowledges the address byte by pulling the data line low. The 8-bit data is then read. The host pulls the data line high (no acknowledge), and a stop condition completes the sequence.

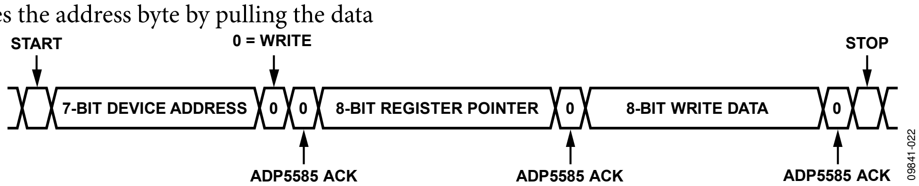


Figure 23. I^{2}C Single Byte Write Sequence

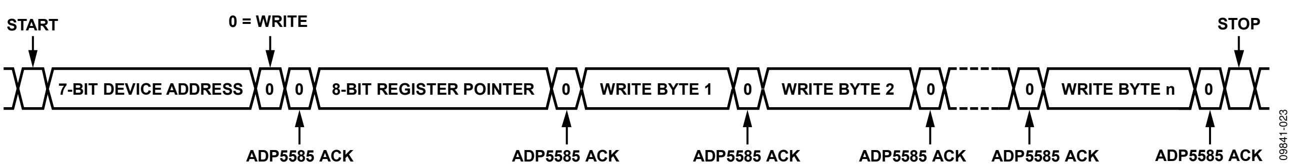


Figure 24. I^{2}C Multibyte Write Sequence

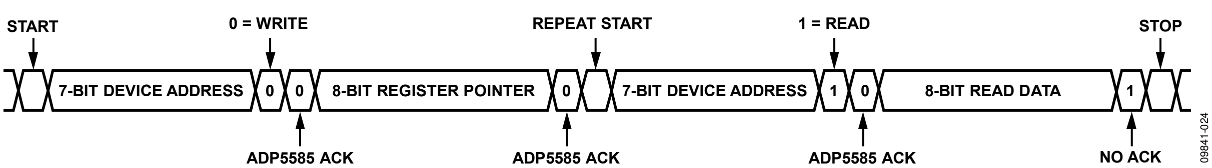


Figure 25. I^{2}C Single Byte Read Sequence

Figure 26 shows a typical multibyte read sequence for reading internal registers. The cycle begins with a start condition, followed by the 7-bit device address (0x34 for all models except the ADP5585ACPZ-03-R7, 0x30 for the ADP5585ACPZ-03-R7 only), followed by the R/$\overline{W}$ bit set to 0 for a write cycle. The ADP5585 acknowledges the address byte by pulling the data line low. The address of the register from which data is to be read is sent next. The ADP5585 acknowledges the register pointer byte by pulling the data line low. A start condition is repeated, followed by the 7-bit device address (0x34 for all models except the ADP5585ACPZ-03-R7, 0x30 for the ADP5585ACPZ-03-R7 only), followed by the R/$\overline{W}$ bit set to 1 for a read cycle. The ADP5585 acknowledges the address byte by pulling the data line low. The 8-bit data is then read. The address pointer is then incremented to read the next data byte, and the host continues to pull the data line low for each byte (master acknowledge) until the n data byte is read. The host pulls the data line high (no acknowledge) after the last byte is read, and a stop condition completes the sequence.

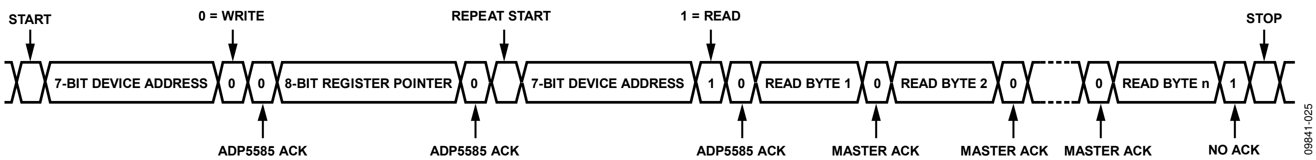


Figure 26. I²C Multibyte Read Sequence

REGISTER MAP

Table 6.

Reg Add	Reg Name	R/W[1]	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x00	ID	R	MAN_ID				REV_ID			
0x01	INT_STATUS	R/W	Reserved			LOGIC_INT	Reserved	OVRFLOW_INT	GPI_INT	EVENT_INT
0x02	Status	R	Reserved	LOGIC_STAT	Reserved	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	Reserved		GPI_6_INT	GPI_5_INT	GPI_4_INT	GPI_3_INT	GPI_2_INT	GPI_1_INT
0x14	GPI_INT_STAT_B	R	Reserved			GPI_11_INT	GPI_10_INT	GPI_9_INT	GPI_8_INT	GPI_7_INT
0x15	GPI_STATUS_A	R	Reserved		GPI_6_STAT	GPI_5_STAT	GPI_4_STAT	GPI_3_STAT	GPI_2_STAT	GPI_1_STAT
0x16	GPI_STATUS_B	R	Reserved			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	Reserved				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	Reserved						C4_PULL_CFG	
0x1B	GPI_INT_LEVEL_A	R/W	Reserved		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	Reserved			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	Reserved		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	Reserved			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	Reserved		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	Reserved			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	Reserved		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	Reserved			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	Reserved		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	Reserved			GPO_11_DATA	GPO_10_DATA	GPO_9_DATA	GPO_8_DATA	GPO_7_DATA
0x25	GPO_OUT_MODE_A	R/W	Reserved		GPO_6_OUT_MODE	GPO_5_OUT_MODE	GPO_4_OUT_MODE	GPO_3_OUT_MODE	GPO_2_OUT_MODE	GPO_1_OUT_MODE

Reg Add	Reg Name	R/W[1]	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x26	GPO_OUT_MODE_B	R/W	Reserved			GPO_11_OUT_MODE	GPO_10_OUT_MODE	GPO_9_OUT_MODE	GPO_8_OUT_MODE	GPO_7_OUT_MODE
0x27	GPIO_DIRECTION_A	R/W	Reserved		GPO_6_DIR	GPO_5_DIR	GPO_4_DIR	GPO_3_DIR	GPO_2_DIR	GPO_1_DIR
0x28	GPIO_DIRECTION_B	R/W	Reserved			GPO_11_DIR	GPO_10_DIR	GPO_9_DIR	GPO_8_DIR	GPO_7_DIR
0x29	RESET1_EVENT_A	R/W	RESET1_EVENT_A_LEVEL	RESET1_EVENT_A [6:0]						
0x2A	RESET1_EVENT_B	R/W	RESET1_EVENT_B_LEVEL	RESET1_EVENT_B [6:0]						
0x2B	RESET1_EVENT_C	R/W	RESET1_EVENT_C_LEVEL	RESET1_EVENT_C [6:0]						
0x2C	RESET2_EVENT_A	R/W	RESET2_EVENT_A_LEVEL	RESET2_EVENT_A [6:0]						
0x2D	RESET2_EVENT_B	R/W	RESET2_EVENT_B_LEVEL	RESET2_EVENT_B [6:0]						
0x2E	RESET2_CFG	R/W	RESET2_POL	RESET1_POL	$\overline{\text{RST}}$_PASSTHRU_EN	RESET_TRIG_TIME[2:0]			RESET_PULSE_WIDTH[1:0]	
0x2F	PWM_OFFT_LOW	R/W	PWM_OFFT_LOW_BYTE[7:0]							
0x30	PWM_OFFT_HIGH	R/W	PWM_OFFT_HIGH_BYTE[7:0]							
0x31	PWM_ONT_LOW	R/W	PWM_ONT_LOW_BYTE[7:0]							
0x32	PWM_ONT_HIGH	R/W	PWM_ONT_HIGH_BYTE[7:0]							
0x33	PWM_CFG	R/W	Reserved					PWM_IN_AND	PWM_MODE	PWM_EN
0x34	LOGIC_CFG	R/W	Reserved	LY_INV	LC_INV	LB_INV	LA_INV	LOGIC_SEL[2:0]		
0x35	LOGIC_FF_CFG	R/W	Reserved						FF_SET	FF_CLR
0x36	LOGIC_INT_EVENT_EN	R/W	Reserved					LY_DBNC_DIS	LOGIC_EVENT_EN	LOGIC_INT_LEVEL
0x37	POLL_TIME_CFG	R/W	Reserved						KEY_POLL_TIME[1:0]	
0x38	PIN_CONFIG_A	R/W	Reserved		R5_CONFIG	R4_CONFIG	R3_CONFIG	R2_CONFIG	R1_CONFIG	R0_CONFIG
0x39	PIN_CONFIG_B	R/W	Reserved			C4_CONFIG	C3_CONFIG	C2_CONFIG	C1_CONFIG	C0_CONFIG
0x3A	PIN_CONFIG_C	R/W	PULL_SELECT	C4_EXTEND_CFG	R4_EXTEND_CFG	Reserved	R3_EXTEND_CFG[1:0]		Reserved	R0_EXTEND_CFG
0x3B	GENERAL_CFG	R/W	OSC_EN	CORE_FREQ[1:0]		Reserved			$\overline{\text{INT}}$_CFG	$\overline{\text{RST}}$_CFG
0x3C	INT_EN	R/W	Reserved			LOGIC_IEN	Reserved	OVRFLOW_IEN	GPI_IEN	EVENT_IEN

[1] R means read, W means write, and R/W means read/write.

DETAILED REGISTER DESCRIPTIONS

Note that N/A throughout this section means not applicable.

Note: All register default to 0000 0000 unless otherwise specified.

ID Register 0x00

Table 7. ID Bit Descriptions

Bit(s)	Bit Name	Access	Description
7 to 4	MAN_ID	Read only	Manufacturer ID, default = 0010
3 to 0	REV_ID	Read only	Rev ID

INT_STATUS Register 0x01

Table 8. INT_STATUS Bit Descriptions

Bit(s)	Bit Name	Access	Description[1]
7 to 5	N/A		Reserved.
4	LOGIC_INT	Read/write	0 = no interrupt. 1 = interrupt due to a general logic condition.
3	N/A		Reserved.
2	OVERFLOW_INT	Read/write	0 = no interrupt. 1 = interrupt due to an overflow condition.
1	GPI_INT	Read/write	This bit is not set by a GPI that has been configured to update the FIFO and event count. This bit cannot be cleared until all GPI_x_INT bits are cleared. 0 = no interrupt. 1 = interrupt due to a general GPI condition.
0	EVENT_INT	Read/write	0 = no interrupt. 1 = interrupt due to key event (press/release), GPI event (GPI programmed for FIFO updates), or logic event (programmed for FIFO updates).

[1] Interrupt bits are cleared by writing a 1 to the flag; writing a 0 or reading the flag has no effect.

Status Register 0x02

Table 9. Status Bit Descriptions

Bit(s)	Bit Name	Access	Description
7	N/A		Reserved.
6	LOGIC_STAT	Read only	0 = output from logic block (LY) is low. 1 = output from logic block (LY) is high.
5	N/A		Reserved.
4 to 0	EC[4:0]	Read only	Event count value. Indicates how many events are currently stored on the FIFO.

FIFO_1 Register 0x03

Table 10. FIFO_1 Bit Descriptions

Bit(s)	Bit Name	Access	Description
7	EVENT1_STATE	Read only	This bit represents the state of the event that is recorded in the EVENT1_IDENTIFIER[6:0] bit. For key events from Event 1 to Event 36, use the following settings: 1 = key is pressed. 0 = key is released. For GPI and logic events from Event 37 to Event 48, use the following settings: 1 = GPI/logic is active. 0 = GPI/logic is inactive. Active and inactive states for Event 37 to Event 48 are programmable.
6 to 0	EVENT1_IDENTIFIER[6:0]	Read only	Contains the event identifier for the pin. Refer to Table 11.

Table 11. Event Decoding

Event No.	Meaning	Event No.	Meaning	Event No.	Meaning	Event No.	Meaning
0	No event	32	Key 32 (R1, GND)	64	Unused	96	Unused
1	Key 1 (R0, C0)	33	Key 33 (R2, GND)	65	Unused	97	Unused
2	Key 2 (R0, C1)	34	Key 34 (R3, GND)	66	Unused	98	Unused
3	Key 3 (R0, C2)	35	Key 35 (R4, GND)	67	Unused	99	Unused
4	Key 4 (R0, C3)	36	Key 36 (R5, GND)	68	Unused	100	Unused
5	Key 5 (R0, C4)	37	GPI 1 (R0)	69	Unused	101	Unused
6	Key 6 (R1, C0)	38	GPI 2 (R1)	70	Unused	102	Unused
7	Key 7 (R1, C1)	39	GPI 3 (R2)	71	Unused	103	Unused
8	Key 8 (R1, C2)	40	GPI 4 (R3)	72	Unused	104	Unused
9	Key 9 (R1, C3)	41	GPI 5 (R4)	73	Unused	105	Unused
10	Key 10 (R1, C4)	42	GPI 6 (R5)	74	Unused	106	Unused
11	Key 11 (R2, C0)	43	GPI 7 (C0)	75	Unused	107	Unused
12	Key 12 (R2, C1)	44	GPI 8 (C1)	76	Unused	108	Unused
13	Key 13 (R2, C2)	45	GPI 9 (C2)	77	Unused	109	Unused
14	Key 14 (R2, C3)	46	GPI 10 (C3)	78	Unused	110	Unused
15	Key 15 (R2, C4)	47	GPI 11 (C4)	79	Unused	111	Unused
16	Key 16 (R3, C0)	48	Logic	80	Unused	112	Unused
17	Key 17 (R3, C1)	49	Unused	81	Unused	113	Unused
18	Key 18 (R3, C2)	50	Unused	82	Unused	114	Unused
19	Key 19 (R3, C3)	51	Unused	83	Unused	115	Unused
20	Key 20 (R3, C4)	52	Unused	84	Unused	116	Unused
21	Key 21 (R4, C0)	53	Unused	85	Unused	117	Unused
22	Key 22 (R4, C1)	54	Unused	86	Unused	118	Unused
23	Key 23 (R4, C2)	55	Unused	87	Unused	119	Unused
24	Key 24 (R4, C3)	56	Unused	88	Unused	120	Unused
25	Key 25 (R4, C4)	57	Unused	89	Unused	121	Unused
26	Key 26 (R5, C0)	58	Unused	90	Unused	122	Unused
27	Key 27 (R5, C1)	59	Unused	91	Unused	123	Unused
28	Key 28 (R5, C2)	60	Unused	92	Unused	124	Unused
29	Key 29 (R5, C3)	61	Unused	93	Unused	125	Unused
30	Key 30 (R5, C4)	62	Unused	94	Unused	126	Unused
31	Key 31 (R0, GND)	63	Unused	95	Unused	127	Unused

FIFO_2 Register 0x04

Table 12. FIFO_2 Bit Descriptions

Bit(s)	Bit Name	Access	Description
7	EVENT2_STATE	Read only	Refer to Table 10.
6 to 0	EVENT2_IDENTIFIER[6:0]	Read only	Refer to Table 10.

FIFO_3 Register 0x05

Table 13. FIFO_3 Bit Descriptions

Bit(s)	Bit Name	Access	Description
7	EVENT3_STATE	Read only	Refer to Table 10.
6 to 0	EVENT3_IDENTIFIER[6:0]	Read only	Refer to Table 10.

FIFO_4 Register 0x06

Table 14. FIFO_4 Bit Descriptions

Bit(s)	Bit Name	Access	Description
7	EVENT4_STATE	Read only	Refer to Table 10.
6 to 0	EVENT4_IDENTIFIER[6:0]	Read only	Refer to Table 10.

FIFO_5 Register 0x07

Table 15. FIFO_5 Bit Descriptions

Bit(s)	Bit Name	Access	Description
7	EVENT5_STATE	Read only	Refer to Table 10.
6 to 0	EVENT5_IDENTIFIER[6:0]	Read only	Refer to Table 10.

FIFO_6 Register 0x08

Table 16. FIFO_6 Bit Descriptions

Bit(s)	Bit Name	Access	Description
7	EVENT6_STATE	Read only	Refer to Table 10.
6 to 0	EVENT6_IDENTIFIER[6:0]	Read only	Refer to Table 10.

FIFO_7 Register 0x09

Table 17. FIFO_7 Bit Descriptions

Bit(s)	Bit Name	Access	Description
7	EVENT7_STATE	Read only	Refer to Table 10.
6 to 0	EVENT7_IDENTIFIER[6:0]	Read only	Refer to Table 10.

FIFO_8 Register 0x0A

Table 18. FIFO_8 Bit Descriptions

Bit(s)	Bit Name	Access	Description
7	EVENT8_STATE	Read only	Refer to Table 10.
6 to 0	EVENT8_IDENTIFIER[6:0]	Read only	Refer to Table 10.

FIFO_9 Register 0x0B

Table 19. FIFO_9 Bit Descriptions

Bit(s)	Bit Name	Access	Description
7	EVENT9_STATE	Read only	Refer to Table 10.
6 to 0	EVENT9_IDENTIFIER[6:0]	Read only	Refer to Table 10.

FIFO_10 Register 0x0C

Table 20. FIFO_10 Bit Descriptions

Bit(s)	Bit Name	Access	Description
7	EVENT10_STATE	Read only	Refer to Table 10.
6 to 0	EVENT10_IDENTIFIER[6:0]	Read only	Refer to Table 10.

FIFO_11 Register 0x0D

Table 21. FIFO_11 Bit Descriptions

Bit(s)	Bit Name	Access	Description
7	EVENT11_STATE	Read only	Refer to Table 10.
6 to 0	EVENT11_IDENTIFIER[6:0]	Read only	Refer to Table 10.

FIFO_12 Register 0x0E

Table 22. FIFO_12 Bit Descriptions

Bit(s)	Bit Name	Access	Description
7	EVENT12_STATE	Read only	Refer to Table 10.
6 to 0	EVENT12_IDENTIFIER[6:0]	Read only	Refer to Table 10.

FIFO_13 Register 0x0F

Table 23. FIFO_13 Bit Descriptions

Bit(s)	Bit Name	Access	Description
7	EVENT13_STATE	Read only	Refer to Table 10.
6 to 0	EVENT13_IDENTIFIER[6:0]	Read only	Refer to Table 10.

FIFO_14 Register 0x10

Table 24. FIFO_14 Bit Descriptions

Bit(s)	Bit Name	Access	Description
7	EVENT14_STATE	Read only	Refer to Table 10.
6 to 0	EVENT14_IDENTIFIER[6:0]	Read only	Refer to Table 10.

FIFO_15 Register 0x11

Table 25. FIFO_15 Bit Descriptions

Bit(s)	Bit Name	Access	Description
7	EVENT15_STATE	Read only	Refer to Table 10.
6 to 0	EVENT15_IDENTIFIER[6:0]	Read only	Refer to Table 10.

FIFO_16 Register 0x12

Table 26. FIFO_16 Bit Descriptions

Bit(s)	Bit Name	Access	Description
7	EVENT16_STATE	Read only	Refer to Table 10.
6 to 0	EVENT16_IDENTIFIER[6:0]	Read only	Refer to Table 10.

GPI_INT_STAT_A Register 0x13

Table 27. GPI_INT_STAT_A Bit Descriptions

Bit(s)	Bit Name	Access	Description
7 to 6	N/A		Reserved.
5	GPI_6_INT	Read only	0 = no interrupt 1 = interrupt due to GPI_6 (R5 pin). Cleared on read.
4	GPI_5_INT	Read only	0 = no interrupt 1 = interrupt due to GPI_5 (R4 pin). Cleared on read.
3	GPI_4_INT	Read only	0 = no interrupt 1 = interrupt due to GPI_4 (R3 pin). Cleared on read.
2	GPI_3_INT	Read only	0 = no interrupt 1 = interrupt due to GPI_3 (R2 pin). Cleared on read.
1	GPI_2_INT	Read only	0 = no interrupt 1 = interrupt due to GPI_2 (R1 pin). Cleared on read.
0	GPI_1_INT	Read only	0 = no interrupt 1 = interrupt due to GPI_1 (R0 pin). Cleared on read.

GPI_INT_STAT_B Register 0x14

Table 28. GPI_INT_STAT_B Bit Descriptions

Bit(s)	Bit Name	Access	Description
7 to 5	N/A		Reserved.
4	GPI_11_INT	Read only	0 = no interrupt. 1 = interrupt due to GPI_11 (C4 pin). Cleared on read.
3	GPI_10_INT	Read only	0 = no interrupt. 1 = interrupt due to GPI_10 (C3 pin). Cleared on read.

Bit(s)	Bit Name	Access	Description
2	GPI_9_INT	Read only	0 = no interrupt. 1 = interrupt due to GPI_9 (C2 pin). Cleared on read.
1	GPI_8_INT	Read only	0 = no interrupt. 1 = interrupt due to GPI_8 (C1 pin). Cleared on read.
0	GPI_7_INT	Read only	0 = no interrupt. 1 = interrupt due to GPI_7 (C0 pin). Cleared on read.

GPI_STATUS_A Register 0x15

Table 29. GPI_STATUS_A Bit Descriptions

Bit(s)	Bit Name	Access	Description
7 to 6	N/A		Reserved.
5	GPI_6_STAT	Read only	0 = GPI_6 (R5 pin) is low. 1 = GPI_6 (R5 pin) is high.
4	GPI_5_STAT	Read only	0 = GPI_5 (R4 pin) is low. 1 = GPI_5 (R4 pin) is high.
3	GPI_4_STAT	Read only	0 = GPI_4 (R3 pin) is low. 1 = GPI_4 (R3 pin) is high.
2	GPI_3_STAT	Read only	0 = GPI_3 (R2 pin) is low. 1 = GPI_3 (R2 pin) is high.
1	GPI_2_STAT	Read only	0 = GPI_2 (R1 pin) is low. 1 = GPI_2 (R1 pin) is high.
0	GPI_1_STAT	Read only	0 = GPI_1 (R0 pin) is low. 1 = GPI_1 (R0 pin) is high.

GPI_STATUS_B Register 0x16

Table 30. Register 0x16, GPI_STATUS_B Bit Descriptions

Bit(s)	Bit Name	Access	Description
7 to 5	N/A		Reserved.
4	GPI_11_STAT	Read only	0 = GPI_11 (C4 pin) is low. 1 = GPI_11 (C4 pin) is high.
3	GPI_10_STAT	Read only	0 = GPI_10 (C3 pin) is low. 1 = GPI_10 (C3 pin) is high.
2	GPI_9_STAT	Read only	0 = GPI_9 (C2 pin) is low. 1 = GPI_9 (C2 pin) is high.
1	GPI_8_STAT	Read only	0 = GPI_8 (C1 pin) is low. 1 = GPI_8 (C1 pin) is high.
0	GPI_7_STAT	Read only	0 = GPI_7 (C0 pin) is low. 1 = GPI_7 (C0 pin) is high.

RPULL_CONFIG_A Register 0x17

Table 31. RPULL_CONFIG_A Bit Descriptions

Bit(s)	Bit Name	Access	Description
7 to 6	R3_PULL_CFG	Read/write	00 = enable 300 kΩ pull-up resistor. 01 = enable 300 kΩ pull-down resistor. 10 = enable 100 kΩ pull-up resistor. 11 = disable all pull-up/pull-down resistors.
5 to 4	R2_PULL_CFG	Read/write	00 = enable 300 kΩ pull-up resistor. 01 = enable 300 kΩ pull-down resistor. 10 = enable 100 kΩ pull-up resistor. 11 = disable all pull-up/pull-down resistors.
3 to 2	R1_PULL_CFG	Read/write	00 = enable 300 kΩ pull-up resistor. 01 = enable 300 kΩ pull-down resistor. 10 = enable 100 kΩ pull-up resistor. 11 = disable all pull-up/pull-down resistors.
1 to 0	R0_PULL_CFG	Read/write	00 = enable 300 kΩ pull-up resistor. 01 = enable 300 kΩ pull-down resistor. 10 = enable 100 kΩ pull-up resistor. 11 = disable all pull-up/pull-down resistors.
ADP5585AC_Z-00-R7, ADP5585AC_Z-01-R7, ADP5585ACPZ-03-R7 Default = 0000 0000 ADP5585ACBZ-02-R7 Default = 1100 0011 ADP5585ACBZ-04-R7 Default = 0101 0101			

RPULL_CONFIG_B Register 0x18

Table 32. RPULL_CONFIG_B Bit Descriptions

Bit(s)	Bit Name	Access	Description
7 to 4	N/A		Reserved.
3 to 2	R5_PULL_CFG	Read/write	(Reserved except for ADP5585ACBZ-01-R7 options) 00 = enable 300 kΩ pull-up resistor. 01 = enable 300 kΩ pull-down resistor. 10 = enable 100 kΩ pull-up resistor. 11 = disable all pull-up/pull-down resistors.
1 to 0	R4_PULL_CFG	Read/write	00 = enable 300 kΩ pull-up resistor. 01 = enable 300 kΩ pull-down resistor. 10 = enable 100 kΩ pull-up resistor. 11 = disable all pull-up/pull-down resistors.
ADP5585AC_Z-00-R7, ADP5585AC_Z-01-R7, ADP5585ACPZ-03-R7 Default = 0000 0000 ADP5585ACBZ-02-R7 Default = 0000 0011 ADP5585ACBZ-04-R7 Default = 0000 0101			