

FEATURES

- 18-GPIO port expander or 10 × 8 keypad matrix**
- GPIOs configurable as GPIs, GPOs, and keypad rows or columns**
- I²C interface with auto-increment**
- 1.65 V to 3.6 V operation**
- Keypad lock capability**
- Open-drain interrupt output**
- Key press and key release interrupts**
- GPI interrupt with level programmability**
- Programmable pull-ups**
- Key event counter with overflow interrupt**
- 275 μs debounce on the reset line and GPIs**
- 1 μA typical idle current**
- 55 μA typical polling current**
- 4 mm × 4 mm LFCSP package**
- Small 2 mm × 2 mm WLCSP package, 0.4 mm pitch**
- Multiple I²C addresses available for the LFCSP package to allow multiple port expanders on the same bus**

APPLICATIONS

Keypad and I/O expander designed for QWERTY type phones that require a large keypad matrix

GENERAL DESCRIPTION

The ADP5587 is an I/O port expander and keypad matrix designed for QWERTY type phones that require a large keypad matrix and expanded I/O lines. I/O expander ICs are used in mobile platforms as a solution to the limited number of GPIOs available in the main processor.

In its small 2 mm × 2 mm package, the ADP5587 contains enough power to handle all key scanning and decoding and to flag the processor of key presses and releases via the I²C interface and interrupt. The ADP5587 frees the main micro-processor from the need to monitor the keypad, thereby minimizing current drain and increasing processor bandwidth. The ADP5587 is also equipped with a buffer/FIFO and key event counter to handle and keep track of up to 10 unprocessed key or GPI events with overflow wrap and interrupt capability.

The ADP5587 has keypad lock capability with an option to trigger or not trigger an interrupt at key presses and releases.

FUNCTIONAL BLOCK DIAGRAM

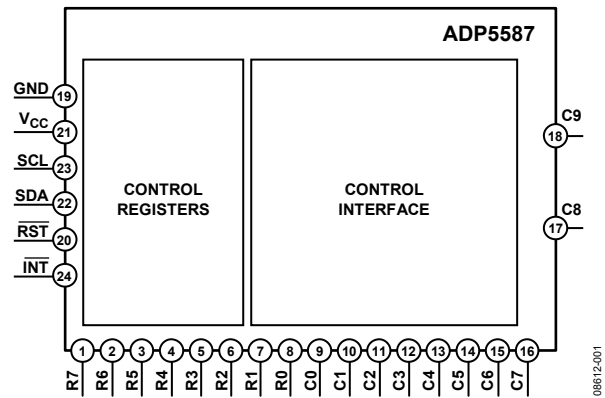


Figure 1.

All communication to the main processor is done using one interrupt line and two I²C-compatible interface lines. The ADP5587 can be configured as a keypad matrix of up to 8 rows × 10 columns (a maximum of 80 keys).

When the ADP5587 is used for smaller keypad matrices, unused row and column pins can be reconfigured to act as general-purpose inputs or outputs. R0 to R7 denote the row pins of the matrix, whereas C0 to C9 denote the column pins. At power-up, all rows and columns default to GPIs and must be programmed to function as part of the keypad matrix or as GPOs.

Two options for I²C addresses exist for the LFCSP package to reduce the chance of port contention and allow up to two ADP5587 ICs to operate on the same I²C bus (see the Ordering Guide).

TABLE OF CONTENTS

| | | | |
|--|---|---|----|
| Features | 1 | Typical Performance Characteristics | 7 |
| Applications..... | 1 | Theory of Operation | 8 |
| Functional Block Diagram | 1 | Keypad Operation | 8 |
| General Description | 1 | General-Purpose Inputs and Outputs | 12 |
| Revision History | 2 | I ² C Programming and Digital Control..... | 14 |
| Specifications..... | 3 | Registers..... | 15 |
| DC Characteristics | 3 | Register Descriptions..... | 16 |
| AC Characteristics..... | 4 | Applications Information | 21 |
| Absolute Maximum Ratings..... | 5 | Applications Overview | 21 |
| Thermal Resistance | 5 | Keypad Current | 21 |
| ESD Caution..... | 5 | Outline Dimensions | 23 |
| Pin Configurations and Function Descriptions | 6 | Ordering Guide | 23 |

REVISION HISTORY

10/2019—Rev. D to Rev. E

| | |
|----------------------------------|----|
| Changes to Table 16..... | 16 |
| Updated Outline Dimensions | 23 |
| Changes to Ordering Guide | 23 |

1/2012—Rev. C to Rev. D

| | |
|--------------------------|----|
| Changes to Table 11..... | 9 |
| Changes to Table 25..... | 18 |

7/2011—Rev. B to Rev. C

| | |
|--|----|
| Changes to Features and General Description, I ² C Address Options | 1 |
| Changes to the I ² C Programming and Digital Control Section, Figure 16, Figure 17, and Figure 18..... | 14 |
| Changes to Ordering Guide | 23 |

5/2010—Rev. A to Rev. B

| | |
|---------------------------|---|
| Changes to Features | 1 |
| Changes to Table 1..... | 3 |
| Changes to Table 8..... | 6 |

3/2010—Rev. 0 to Rev. A

| | |
|--|------------|
| Added WLCSP Information | Throughout |
| Added Typical Performance Characteristics Section | 7 |
| Updated Outline Dimensions, Changes to Ordering Guide | 23 |

12/2009—Revision 0: Initial Version

SPECIFICATIONS

$T_A = T_J = -40^\circ\text{C}$ to $+85^\circ\text{C}$, unless otherwise noted.

DC CHARACTERISTICS

Table 1. General DC Electrical Characteristics

| Parameter | Symbol | Conditions | Min | Typ | Max | Unit |
|---|----------|---|------|-----|-----|---------------|
| SUPPLY VOLTAGE | | | | | | |
| V_{CC} Input Voltage Range | V_{CC} | | 1.65 | | 3.6 | V |
| Supply Current ¹ | I_{CC} | $V_{CC} = 1.8\text{ V to }3.0\text{ V}, T_A = -40^\circ\text{C to }+85^\circ\text{C}$ | | 1 | 10 | μA |
| With One Key Press | I_{CC} | $V_{CC} = 1.8\text{ V}, T_A = -40^\circ\text{C to }+85^\circ\text{C}$ | | 55 | 90 | μA |
| With GPI Low (Pull-Up Enabled) ² | I_{CC} | $V_{CC} = 3.0\text{ V}, T_A = -40^\circ\text{C to }+85^\circ\text{C}$ | | 100 | 200 | μA |
| With GPI Low (Pull-Up Disabled) | I_{CC} | $V_{CC} = 1.8\text{ V to }3.0\text{ V}, T_A = -40^\circ\text{C to }+85^\circ\text{C}$ | | 2 | 10 | μA |
| With One GPO Active ³ | I_{CC} | $V_{CC} = 1.8\text{ V}, T_A = -40^\circ\text{C to }+85^\circ\text{C}$ | | | 50 | μA |
| OSCILLATOR CURRENT | | | | | | |
| Oscillator Current (Enabled) | I_{CC} | $V_{CC} = 1.8\text{ V to }3.0\text{ V}$ | | 40 | | μA |

¹ Operating current measured with I/Os defaulting as GPIs, with all pull-ups enabled and all inputs open.

² With one GPI low.

³ Load = 100 k Ω .

Table 2. I/O DC Electrical Characteristics

| Parameter | Symbol | Conditions | Min | Typ | Max | Unit |
|---|-----------------|---|---------------------|-------------------------|---------------------|---------------|
| INPUT LOGIC LEVELS (SCL, SDA, RST, C0 to C9, R0 to R7) ¹ | | | | | | |
| Logic Low Input Voltage | V_{IL} | $1.8\text{ V} \leq V_{IO} \leq 3.0\text{ V}$ | | | $0.3 \times V_{CC}$ | V |
| Logic High Input Voltage | V_{IH} | $1.8\text{ V} \leq V_{IO} \leq 3.0\text{ V}$ | $0.7 \times V_{CC}$ | | | V |
| Schmitt Trigger Hysteresis | V_{HYST} | | | 0.10 | | V |
| Input Leakage Current | $V_{I-LEAKAGE}$ | $1.8\text{ V} \leq V_{IO} \leq 3.0\text{ V}$ | -1 | | +1 | μA |
| OUTPUT LOGIC LEVELS (C0 to C9, R0 to R7) | | | | | | |
| Logic Low Output Voltage | V_{OL} | $I_{SINK} = 1\text{ mA}$ | | | 0.40 | V |
| Output High Voltage | V_{OH} | $I_{SOURCE} = 1\text{ mA}$ | | $V_{CC} - 0.3\text{ V}$ | | V |
| OUTPUT LOGIC LEVELS (INT, SDA) | | | | | | |
| Output Low Voltage | V_{OL} | $I_{SINK} = 3\text{ mA},$ $1.8\text{ V} \leq V_{CC} \leq 3.0\text{ V}$ | | | 0.40 | V |
| Output High Voltage | V_{OH} | $1.8\text{ V} \leq V_{CC} \leq 3.0\text{ V}$ | $0.7 \times V_{CC}$ | | | V |
| Logic High Leakage Current | $V_{O-LEAKAGE}$ | $1.8\text{ V} \leq V_{CC} \leq 3.0\text{ V}$ | | 0.1 | 1 | μA |
| PULL-UP RESISTANCE FOR GPIOs (C0 to C9, R0 to R7) ² | $R_{PULL-UP}$ | | | 100 | | k Ω |

¹ Power-up default current. All I/Os default to GPIs and are open; C8 and C9 default to GPIs; I²C is idle.

² GPIO internal pull-ups are approximately 100 k Ω .

Table 3. Capacitance Loading¹

| Parameter | Symbol | Min | Typ | Max | Unit |
|-----------------------------------|-----------|-----|-----|-----|------|
| I/O Input Capacitance | C_{IN} | | 1 | 10 | pF |
| I/O Output Loading Capacitance | C_{OUT} | | | 50 | pF |
| Capacitive Load for Each Bus Line | C_B^2 | | | 400 | pF |

¹ Guaranteed by design.

² C_B = total capacitance of one bus line in picofarads.

AC CHARACTERISTICS

Table 4. General AC Characteristics¹

| Parameter | Symbol | Min | Typ | Max | Unit |
|---|-------------------|-----|-----|-----|------|
| Delay from Reset Deassertion to I ² C Access | R _{STD} | 60 | | | μs |
| Keypad Unlock Timer | T _{KUT} | | 7 | | sec |
| Keypad Interrupt Mask Timer | T _{KIMT} | | 31 | | sec |
| Debounce | T _D | | 275 | | μs |

¹ Guaranteed by design.

Table 5. I²C AC Electrical Characteristics¹

| Parameter | Symbol | Min | Typ | Max | Unit |
|--|----------------------|-------------------------|-----|-----|------|
| SCL Clock Frequency | f _{SCL} | | | 400 | KHz |
| SCL High Time | t _{HIGH} | 0.6 | | | μs |
| SCL Low Time | t _{LOW} | 1.3 | | | μs |
| Data Setup Time | t _{SU, DAT} | 100 | | | ns |
| Data Hold Time | t _{HD, DAT} | 0 | | 0.9 | μs |
| Setup Time for Repeated Start | t _{SU, STA} | 0.6 | | | μs |
| Hold Time for Start/Repeated Start | t _{HD, STA} | 0.6 | | | μs |
| Bus Free Time for Stop and Start | t _{BUF} | 1.3 | | | μs |
| Setup Time for Stop Condition | t _{SU, STO} | 0.6 | | | μs |
| Rise Time for SCL and SDA ² | t _R | 20 + 0.1 C _B | | 300 | ns |
| Fall Time for SCL and SDA ² | t _F | 20 + 0.1 C _B | | 300 | ns |
| Pulse Width of Suppressed Spike | t _{SP} | 0 | | 50 | μs |

¹ Guaranteed by design.

² t_R and t_F are measured between 0.3 × V_{CC} and 0.7 × V_{CC}.

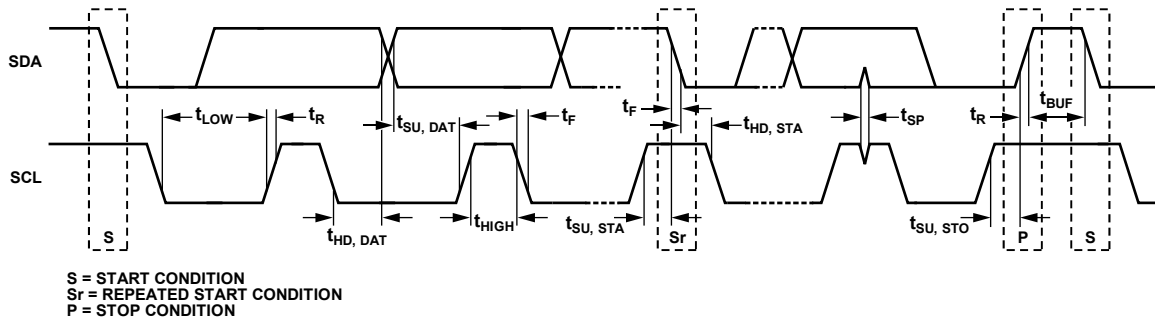


Figure 2. I²C Interface Timing Diagram

08612-002

ABSOLUTE MAXIMUM RATINGS

Table 6.

| Parameter | Rating |
|--------------------------------------|-----------------------------------|
| V _{CC} | −0.3 V to +4.0 V |
| R0 to R7, C0 to C9 | −0.3 V to V _{CC} + 0.3 V |
| SCL | −0.3 V to V _{CC} + 0.3 V |
| SDA | −0.3 V to V _{CC} + 0.3 V |
| RST | −0.3 V to V _{CC} + 0.3 V |
| INT | −0.3 V to V _{CC} + 0.3 V |
| GND | −0.3 V to +0.3 V |
| Operating Ambient Temperature Range | −40°C to +85°C |
| Operating Junction Temperature Range | −40°C to +125°C |
| Storage Temperature Range | −65°C to +150°C |
| ESD Machine Model | ±200 V |
| ESD Human Body Model | ±2000 V |
| ESD Charged Device Model | ±1000 V |
| Soldering Condition | JEDEC J-STD-020 |

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

THERMAL RESISTANCE

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

Table 7. Thermal Resistance

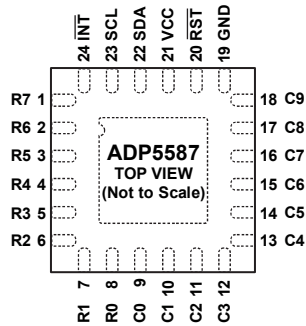
| Package Type | θ_{JA} | θ_{JC} | Unit |
|------------------|---------------|---------------|------|
| 24-Lead LFCSP_WQ | 57.8 | 9.4 | °C/W |
| Maximum Power | 600 | N/A | mW |
| 25-Ball WLCSP | 46 | N/A | °C/W |
| Maximum Power | 600 | N/A | mW |

ESD CAUTION



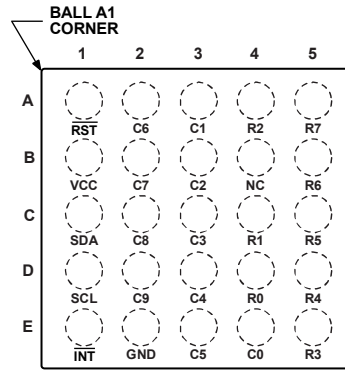
ESD (electrostatic discharge) sensitive device. Charged devices and circuit boards can discharge without detection. Although this product features patented or proprietary protection circuitry, damage may occur on devices subjected to high energy ESD. Therefore, proper ESD precautions should be taken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

PIN CONFIGURATIONS AND FUNCTION DESCRIPTIONS



NOTES
1. EXPOSED PAD. THE EXPOSED PAD MUST BE CONNECTED TO GROUND.

Figure 3. LFCSP Pin Configuration



NOTES
1. NC = NO CONNECT.

Figure 4. WLCSP Pin Configuration

Table 8. Pin Function Descriptions

| LFCSP Pin No. | WLCSP Pin No. | Mnemonic | Description |
|---------------|---------------|-----------------|---|
| 1 | A5 | R7 | GPIO, Row 7 in the Keypad Matrix. |
| 2 | B5 | R6 | GPIO, Row 6 in the Keypad Matrix. |
| 3 | C5 | R5 | GPIO, Row 5 in the Keypad Matrix. |
| 4 | D5 | R4 | GPIO, Row 4 in the Keypad Matrix. |
| 5 | E5 | R3 | GPIO, Row 3 in the Keypad Matrix. |
| 6 | A4 | R2 | GPIO, Row 2 in the Keypad Matrix. |
| N/A | B4 | N/A | No Connect (NC) |
| 7 | C4 | R1 | GPIO, Row 1 in the Keypad Matrix. |
| 8 | D4 | R0 | GPIO, Row 0 in the Keypad Matrix. |
| 9 | E4 | C0 | GPIO, Column 0 in the Keypad Matrix. |
| 10 | A3 | C1 | GPIO, Column 1 in the Keypad Matrix. |
| 11 | B3 | C2 | GPIO, Column 2 in the Keypad Matrix. |
| 12 | C3 | C3 | GPIO, Column 3 in the Keypad Matrix. |
| 13 | D3 | C4 | GPIO, Column 4 in the Keypad Matrix. |
| 14 | E3 | C5 | GPIO, Column 5 in the Keypad Matrix. |
| 15 | A2 | C6 | GPIO, Column 6 in the Keypad Matrix. |
| 16 | B2 | C7 | GPIO, Column 7 in the Keypad Matrix. |
| 17 | C2 | C8 | GPIO, Column 8 in the Keypad Matrix. |
| 18 | D2 | C9 | GPIO, Column 9 in the Keypad Matrix. |
| 19 | E2 | GND | Ground. |
| 20 | A1 | RST | Hardware Reset (Active Low). This pin resets the device to the power default conditions. The reset pin must be driven low for a minimum of 50 μ s to be valid and to prevent false resets due to ESD glitches or noise in the system. If not used, RST must be tied high with a pull-up resistor. |
| 21 | B1 | V _{CC} | Supply Voltage, 1.65 V to 3.6 V. |
| 22 | C1 | SDA | I ² C Serial Data. The open drain requires an external pull-up resistor. |
| 23 | D1 | SCL | I ² C Clock. |
| 24 | E1 | INT | Processor Interrupt, Active Low, Open Drain. This pin can be pulled up to 2.7 V or 1.8 V for selection flexibility in the processor GPIO supply group. |
| EP | N/A | EPAD | Exposed Pad. The exposed pad must be connected to ground. |

TYPICAL PERFORMANCE CHARACTERISTICS

T_A = 25°C, unless otherwise specified.

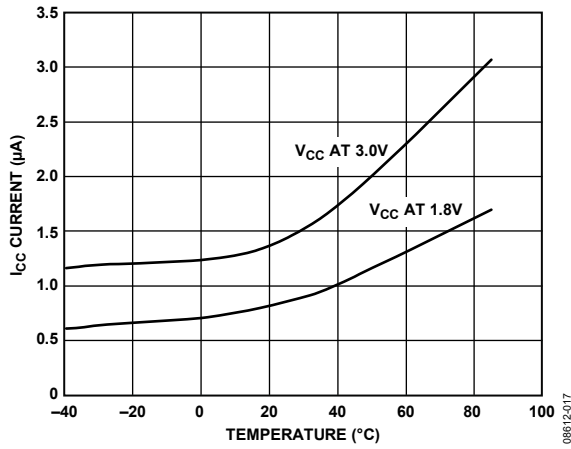


Figure 5. Standby (I_{CC}) Current vs. Temperature

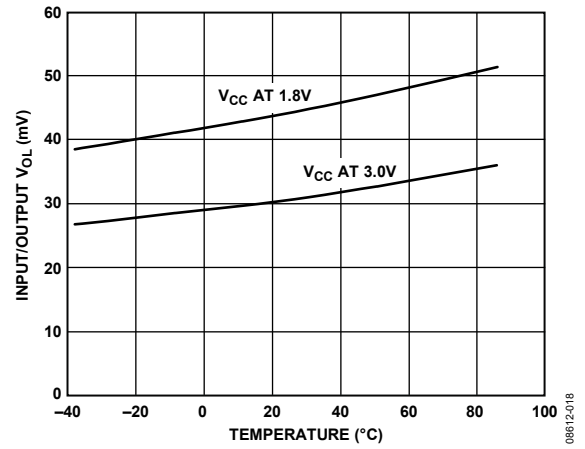


Figure 7. Input/Output V_{OL} vs. Temperature (Sink Current = 1 mA)

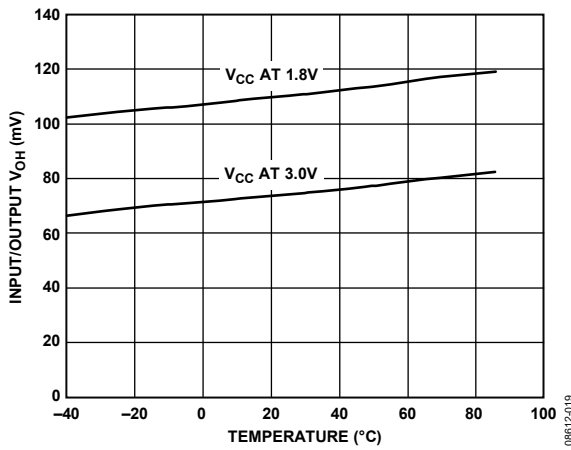


Figure 6. Input/Output V_{OH} vs. Temperature (Source Current = 1 mA)

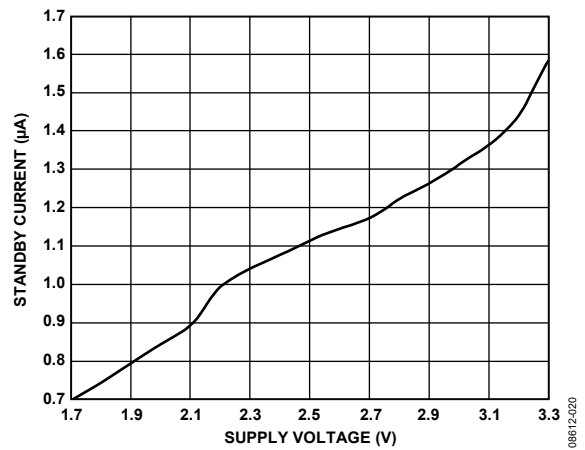


Figure 8. Supply Voltage vs. Standby Current

THEORY OF OPERATION

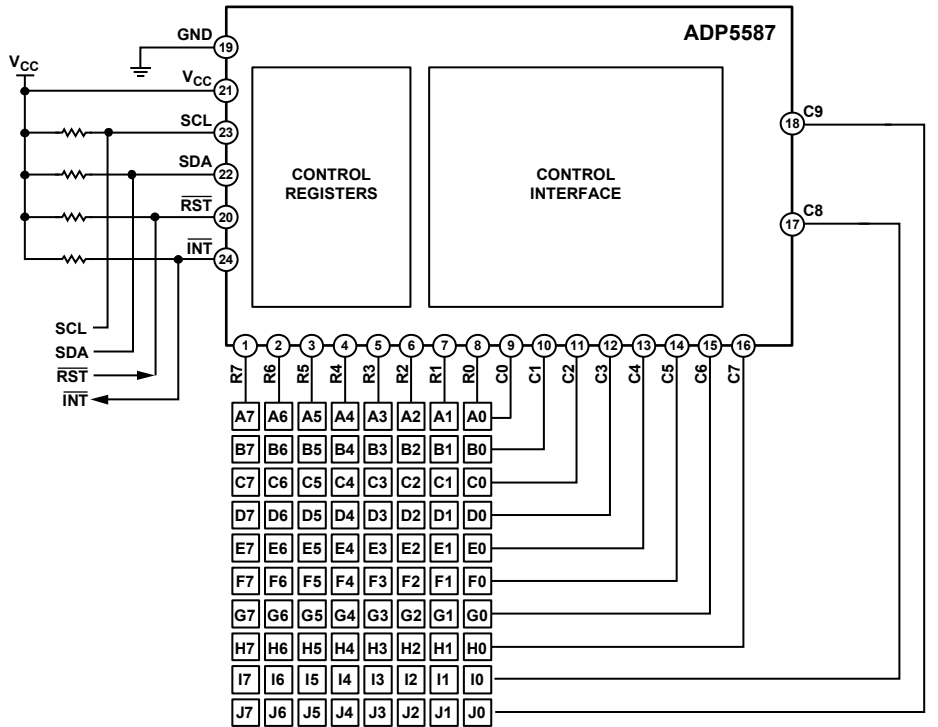


Figure 9. Typical Operating Circuit

The ADP5587 is a GPIO expander that can be configured either as an 18 I/O port expander or as a 10 column × 8 row keypad matrix (80 keys maximum). It is ideal for cellular phone designs and other portable devices that require a large extended keypad and/or expanded I/Os. When smaller size keypads are required, unused GPIOs in the keypad matrix can be used as I/Os (GPOs and GPIs). All GPIOs (rows and columns) default to GPIs at power-up with pull-ups and debounce enabled.

KEYPAD OPERATION

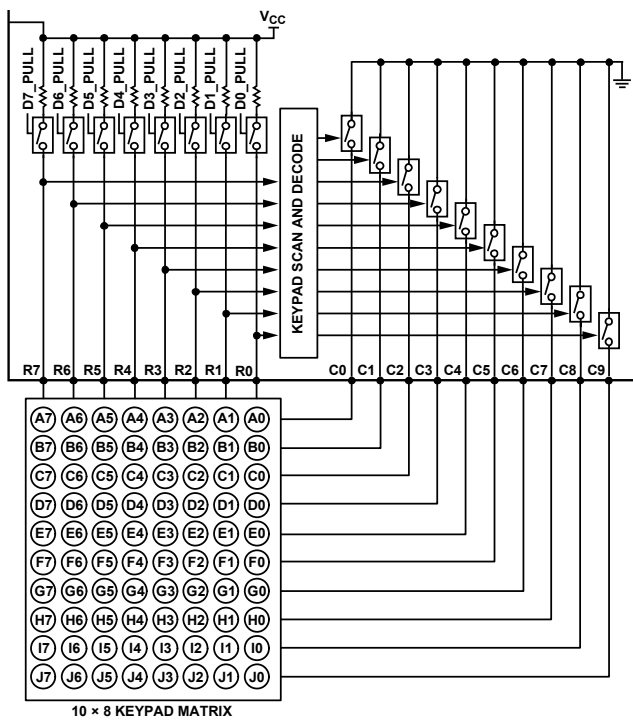
Any number of rows and columns, up to 10 columns × 8 rows, can be configured to be part of the keypad matrix. The rows and columns that make up the keypad matrix must be configured by setting the corresponding bits in Register 0x1D to Register 0x1F. Key presses and releases appear in the key event table/registers with a decimal value of 1 (0x01 hexadecimal or 0000001 binary) through a decimal value of 80 (0x50 hexadecimal or 1010000 binary). See Table 9 for key event number assignments. The keypad, in idle mode, is configured with columns driven low and rows as inputs configured high with pull-up resistors.

Table 9. Key Event Number Assignment Table

| Row | C0 | C1 | C2 | C3 | C4 | C5 | C6 | C7 | C8 | C9 |
|-----|----|----|----|----|----|----|----|----|----|----|
| R0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| R1 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
| R2 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 |
| R3 | 31 | 32 | 33 | 34 | 35 | 36 | 37 | 38 | 39 | 40 |
| R4 | 41 | 42 | 43 | 44 | 45 | 46 | 47 | 48 | 49 | 50 |
| R5 | 51 | 52 | 53 | 54 | 55 | 56 | 57 | 58 | 59 | 60 |
| R6 | 61 | 62 | 63 | 64 | 65 | 66 | 67 | 68 | 69 | 70 |
| R7 | 71 | 72 | 73 | 74 | 75 | 76 | 77 | 78 | 79 | 80 |

When one key press or multiple key presses (short between column and row) occur, the internal state machine checks the row pins to determine which one is driven low and then triggers an internal interrupt. The state machine then starts a key scan cycle to determine which columns are involved in the key press. After a key has been pressed for 25 ms, the state machine sets the appropriate key number in the key event status register with the key-pressed bits set (the MSB in the key event register) in the order detected. The state machine then sets the KE_INT bit in Register 0x02. If the KE_IEN field in Register 0x01 is set, an interrupt is sent to the host processor.

To prevent glitches or narrow press times registering as valid key presses, the key scanner requires the key to be pressed for two scan cycles. The key scanner has a sampling period of 25 ms; therefore, the key must be pressed and held for at least 25 ms to register as pressed. If the key is continuously pressed, the key scanner continues to sample every 25 ms. If a pressed key is released for 25 ms or greater, the state machine sets the appropriate key number in the key event status register with the key-pressed bits cleared in the order detected. Because the release of a key is not necessarily in sync with the key scan sampling period, it may take between 25 ms and 50 ms for a key to register as released. After the key is registered as released, the key scanner returns to idle mode. Figure 10 shows the row and column pins connected to a typical 10 × 8, 80-switch keypad matrix.



NOTES:
1. Dx_PULL STANDS FOR GPIO PULL-UP.

Figure 10. Keypad Decode Configuration

Key Event Tracking

The 10 key event registers are set to act as a FIFO, meaning that reading any of the 10 key event registers yields the key events in the order the keys were pressed and released.

Tracking of key events is done with the help of the key event counter (the KEC field in Register 0x03) and the FIFO/key event registers (Register 0x04 through Register 0x0D). The KEC count increases as keys are pressed and released; up to 10 events can be logged in the counter. The FIFO/key event registers, on the other hand, display the key events and their status (pressed or released) as they are read out of the FIFO. The FIFO registers contain eight bits, with the MSB dedicated as the status bit (1 indicates a press and 0 indicates a release); the remaining seven

bits display the binary representation of the keys that are pressed or released.

The first read of any of the FIFO registers displays the first event that happened and its status. Subsequent reads of the same register replace the register data with the next event that happens. If tracking of all the events is important, it is best to use a single register per event. After all the events in the FIFO are read, reading of any of the event registers yields a zero value.

Table 10 and Table 11 show the event sequences as they are logged in and read from the FIFO. The 10 FIFO registers are labeled A through J, and the keys are labeled A0 through J7.

Table 10. Example of Event Sequence

| Key Pressed/Released | Status | Key Event Counter |
|----------------------|----------|-------------------|
| A0 | Pressed | 1 |
| B1 | Pressed | 2 |
| A0 | Released | 3 |
| C2 | Pressed | 4 |
| B1 | Released | 5 |
| D3 | Pressed | 6 |
| C2 | Released | 7 |
| E4 | Pressed | 8 |
| E4 | Released | 9 |
| D3 | Released | 10 |

Table 11. Interpretation of FIFO Event Reading

| Key Event Counter | Key Event Register Read | Key Event Register Content (Binary) ¹ | Key Event Register Interpretation |
|-------------------|-------------------------|--|-----------------------------------|
| 10 | N/A | N/A | N/A |
| 9 | D | 1 0000001 | Key A0 pressed |
| 8 | E | 1 0001100 | Key B1 pressed |
| 7 | C | 0 0000001 | Key A0 released |
| 6 | F | 1 0010111 | Key C2 pressed |
| 5 | G | 0 0001100 | Key B1 released |
| 4 | A | 1 0100010 | Key D3 pressed |
| 3 | B | 0 0010111 | Key C2 released |
| 2 | H | 1 0101101 | Key E4 pressed |
| 1 | J | 0 0101101 | Key E4 released |
| 0 | I | 0 0100010 | Key D3 released |

¹ The MSB indicates a key press or key release in the key event register: 1 = key press; 0 = key release.

Key Event Overflow

The ADP5587 is equipped with an overflow feature to handle key events beyond the FIFO capacity. When all events are filled, any additional events set the OVR_FLOW_INT bit in Register 0x02; if the OVR_FLOW_IEN bit in Register 0x01 is set, the host processor is also interrupted when overflow occurs. When the FIFO is not full, new events are added as the last events.

The OVR_FLOW_M bit in Register 0x01 sets the mode of operation during overflows. Clearing the OVR_FLOW_M bit causes new incoming events to be discarded, and setting this bit rolls over and overwrites old data with new data starting at the first event.

Auto-Increment

The ADP5587 features automatic increment during I²C read access, which allows the user to increment the address pointer without the need to send a read command for subsequent addresses. This minimizes processor intervention and, therefore, saves processor bandwidth and current drain. Bit 7 of Register 0x01 must be set to initiate auto-increment (see Figure 17 for the full write and read sequence).

Key Event Interrupt

On a key event (KE) interrupt, the processor reads the interrupt status register to determine the cause of the interrupt. If the KE_INT bit in Register 0x02 is set, the processor reads the key event count from the KEC [3:0] field in Register 0x03 to determine the number of events. After reading all the events from the

FIFO, it then reads the KEC field again (in Register 0x03) to make sure that no new events have come in. After all the events are read, the KEC field is decremented to zero (KEC = 0), and the KE_INT bit can be cleared by writing a 1 to it. Both key presses and key releases are capable of generating key event interrupts. The KE_INT bit cannot be cleared, and the INT pin cannot be deasserted, until the FIFO is cleared of all events.

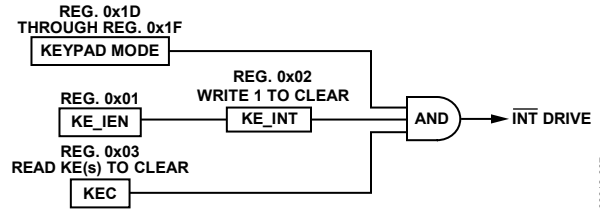


Figure 11. Key Event Interrupt Generation

08612-007

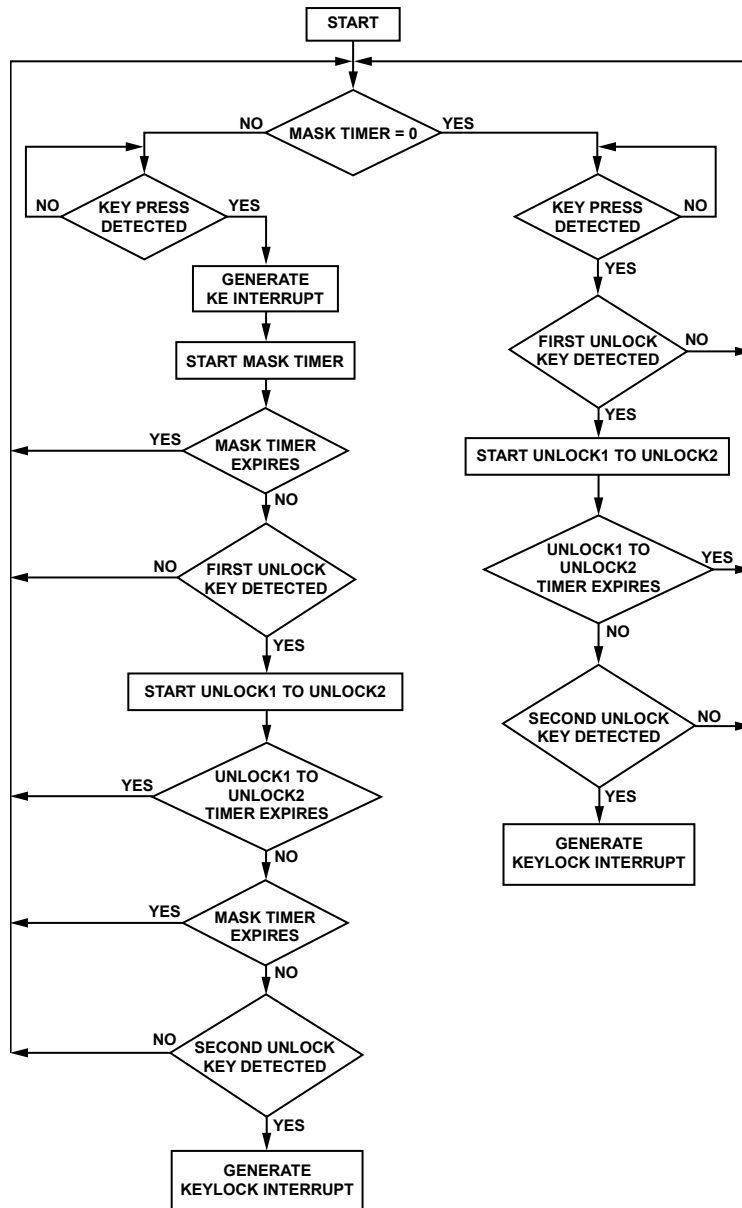


Figure 12. Keypad Lock Interrupt Mask Timer Flowchart

08612-008

Keypad Lock/Unlock Feature

The ADP5587 has a locking feature that allows the user to lock the keypad or GPIs (configured to be part of the event table). When enabled, the keypad lock can prevent generation of key event interrupts and prevent key events from being recorded in the key event table. This feature comprises the Unlock Key 1 and Unlock Key 2 registers (Register 0x0F and Register 0x10, respectively), the keypad lock interrupt mask and keypad unlock timers (Register 0x0E), and the LCK1, LCK2, and keylock enable (K_LCK_EN) bits (Register 0x03).

The unlock keys can be programmed with any value of the keys in the keypad matrix or any GPI event values that are part of the key event table. When the keypad lock interrupt mask timer is enabled, the user must press two specific keys before a keylock interrupt is generated or keypad events are recorded. After the keypad is locked (set Bit 6, Register 0x03, to enable the lock), the first time that the user presses any key, a key event interrupt is generated. No additional interrupt is generated unless both unlock key sequences are correct.

If the correct unlock keys are not pressed before the mask timer expires, the state machine starts over. The first key event interrupt is generated to allow the software to see that the user has pressed a key so that the host can turn on the LCD and display the unlock message. The host then reads the lock status register to see if the keypad is unlocked. After the first key event

interrupt, the state machine does not interrupt the processor again unless the correct sequence is keyed. The state machine is reset if the correct sequences are not keyed before the keypad lock interrupt mask timer expires.

The state of the keypad lock interrupt mask bit (Register 0x01, Bit 2) in the configuration register determines whether the interrupt pin is asserted when the keylock interrupt status bit (Register 0x02, Bit 2) is set. Setting the keylock interrupt mask bit causes the $\overline{\text{INT}}$ pin to be asserted when the keylock interrupt status bit is set in Register 0x02; clearing that bit masks the interrupt, causing the interrupt pin not to respond to the keylock interrupt status bit. The mask interrupt timer should be set for the time that it takes for the LCD to dim or turn off so that, if a key is pressed, the backlight is set to bright mode again or reset to turn on the LCD.

When the unlock mask interrupt timer equals 0, only the correct unlock sequence can generate an interrupt. Disabling the unlock mask interrupt timer allows the processor to remain undisturbed for situations in which the user, for example, has the phone in a pocket or purse and the keys are constantly pressed. The flowchart in Figure 11 shows the interaction of interrupt enable, key event counter, key event interrupt status, and interrupt generation.

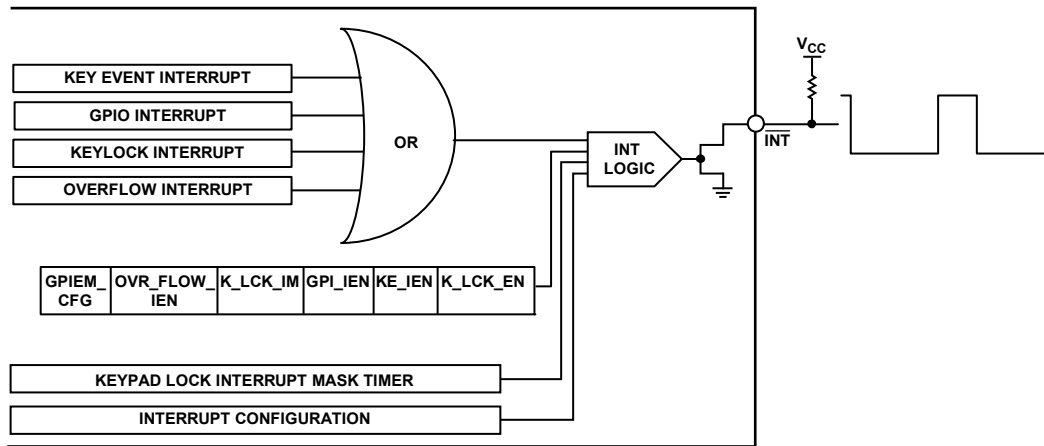
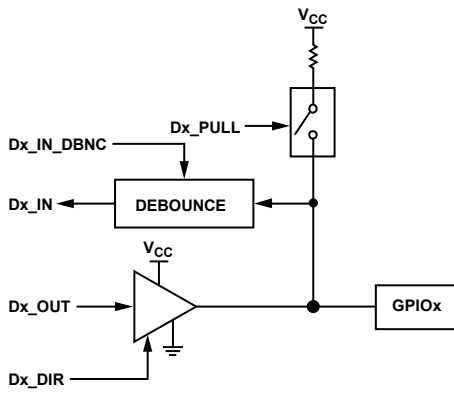


Figure 13. $\overline{\text{INT}}$ Pin Drive

08612-009

GENERAL-PURPOSE INPUTS AND OUTPUTS

The ADP5587 supports up to 18 programmable GPIOs that can be configured to address a variety of uses. Figure 14 shows the makeup of a typical GPIO block where GPIOx represents any of the 18 I/O lines.



- NOTES:
1. Dx_IN STANDS FOR ANY OF THE 18 GPIOs CONFIGURED AS GPIs.
 2. Dx_OUT STANDS FOR ANY OF THE 18 GPIOs CONFIGURED AS GPOS.
 3. Dx_IN_DBNC STANDS FOR GPI DEBOUNCE.
 4. Dx_DIR STANDS FOR GPIO DIRECTION.
 5. Dx_PULL STANDS FOR GPIO PULL-UP.

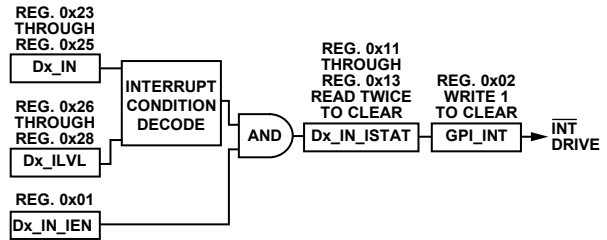
Figure 14. Typical GPIO Block

General-Purpose Inputs (GPI)

The ADP5587 allows the user to configure all or some of its GPIOs as general-purpose inputs (GPIs). After the GPIOs are configured as GPIs, the user can choose to also turn on pull-up resistors and interrupt generation capability, thus reducing the amount of software monitoring and processor interaction and saving power.

The programmed level of the GPI interrupt determines the active level of the GPI pin. For example, if a GPI interrupt level is programmed as high, a high on that pin is considered active and meets the interrupt requirement. If the interrupt is programmed as low, a low on that pin is considered active and meets the interrupt requirement.

GPI data status and interrupt status are reflected in the GPIO interrupt status and data status registers (Register 0x11 through Register 0x16). Caution is necessary during software implementation because an interrupt may be set immediately after the registers are set. To prevent this, the correct logic levels must be present at the GPIs, and the GPIO interrupt level must be set before GPIO interrupt enable or GPI event FIFO enable registers are set. Figure 15 shows the interrupt generation scheme, where Dx represents any one of the 18 GPIOs.



- NOTES:
1. Dx_IN STANDS FOR ANY OF THE 18 GPIOs CONFIGURED AS GPIs.
 2. Dx_ILVL STANDS FOR GPIO INTERRUPT LEVEL.
 3. Dx_IN_IEN STANDS FOR GPI INTERRUPT ENABLE.
 4. Dx_IN_STAT STANDS FOR GPI INTERRUPT STATUS.
 5. GPI_INT STANDS FOR GPI INTERRUPT.

Figure 15. GPIO Interrupt Generation

GPI Events

A column or row configured as a GPI can be programmed to be part of the key event table and is, therefore, also capable of generating a key event interrupt. A key event interrupt caused by a GPI follows the same process flow as a key event interrupt caused by a key press or key release. GPIs configured as part of the key event table allow single key switches and other GPI interrupts to be monitored. As part of the event table, GPIs are represented by a decimal value of 97 (0x61 hexadecimal or 1100001 binary) through a decimal value of 114 (0x72 hexadecimal or 1110010 binary). See Table 12 and Table 13 for GPI event number assignments for rows and columns, respectively.

Table 12. GPI Event Number Assignments for Rows

| R0 | R1 | R2 | R3 | R4 | R5 | R6 | R7 |
|----|----|----|-----|-----|-----|-----|-----|
| 97 | 98 | 99 | 100 | 101 | 102 | 103 | 104 |

Table 13. GPI Event Number Assignments for Columns

| C0 | C1 | C2 | C3 | C4 | C5 | C6 | C7 | C8 | C9 |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| 105 | 106 | 107 | 108 | 109 | 110 | 111 | 112 | 113 | 114 |

For a GPI that is set as active high and is enabled in the key event table, the state machine adds an event to the event count and event tables whenever that GPI goes high. If the GPI is set to active low, a transition from high to low is considered a press and is also added to the event count and event table. After the interrupt state is met, the state machine internally sets an interrupt for the opposite state programmed in the register to prevent polling for the released state, thereby saving current. After the released state is achieved, it is added to the event table. The press and release are still indicated by Bit 7 in the event register (Register 0x04 through Register 0x0D). The GPI events can also be used as unlocked sequences.

When the GPI_EM_REGx bit in Register 0x20 through Register 0x22 is set, GPI events are not tracked when the keypad is locked. The GPIEM_CFG bit (Register 0x01, Bit 6) must be cleared for the GPI events to be tracked in the event counter and event table when the keypad is locked.

275 Microsecond Interrupt Configuration

The ADP5587 gives the user the flexibility of deasserting the interrupt for 275 μ s while there is a pending event. When the INT_CFG bit in Register 0x01 is set, any attempt to clear the interrupt bit while the interrupt pin is already asserted results in a 275 μ s deassertion. When the INT_CFG bit is cleared, the processor interrupt remains asserted if the host tries to clear the interrupt. This feature is particularly useful for software development and edge triggering applications.

Debouncing

The ADP5587 has a 275 μ s debounce time for GPIOs configured as GPIs and rows in keypad scanning mode. The reset line always has a 275 μ s debounce time.

General-Purpose Outputs (GPOs)

The ADP5587 allows the user to configure all or some of its GPIOs as GPOs. These GPOs can be used as extra enables for the host processor or simply as trigger outputs. When configured as an output (GPO), a digital buffer drives the pin to 0 V for a 0 and to V_{CC} for a 1. To set any GPIO as a GPO, make sure that the corresponding bits in Register 0x1D through Register 0x1F are set for GPIO mode; then use Register 0x23 through Register 0x25 to set the corresponding bits for GPO mode.

Power-On Reset

For built-in power-up initialization for applications lacking a power-on reset signal, a reset pin, \overline{RST} , allows the user to reset the registers to default values in the event of a brownout or other reset condition.

Table 14. Device Configuration

| Keypad | | | GPIO | |
|---------------|--------------------|----------------|--------------------|-----------------|
| Matrix | Active Pins | Number of Keys | Available GPIO | Number of GPIOs |
| 10 \times 8 | C0 to C9, R0 to R7 | 80 | 0 | 0 |
| 8 \times 8 | C0 to C7, R0 to R7 | 64 | C8, C9 | 2 |
| 8 \times 7 | C0 to C7, R0 to R6 | 56 | R7, C8, C9 | 3 |
| 8 \times 6 | C0 to C7, R0 to R5 | 48 | R6, R7, C8, C9 | 4 |
| 8 \times 5 | C0 to C7, R0 to R4 | 40 | R5 to R7, C8, C9 | 5 |
| 7 \times 7 | C0 to C6, R0 to R6 | 49 | R7, C7 to C9 | 4 |
| 7 \times 6 | C0 to C6, R0 to R5 | 42 | R6, R7, C7 to C9 | 5 |
| 7 \times 5 | C0 to C6, R0 to R4 | 35 | R5 to R7, C7 to C9 | 6 |
| 6 \times 6 | C0 to C5, R0 to R5 | 36 | R6, R7, C6 to C9 | 6 |
| 6 \times 5 | C0 to C5, R0 to R4 | 30 | R5 to R7, C6 to C9 | 7 |
| 6 \times 4 | C0 to C5, R0 to R3 | 24 | R4 to R7, C6 to C9 | 8 |
| ... | ... | ... | ... | ... |
| 0 \times 0 | None | 0 | R0 to R7, C0 to C9 | 18 |

I²C PROGRAMMING AND DIGITAL CONTROL

The ADP5587 provides full I²C software programmability to facilitate its adoption in various product architectures. All register programming is done via the I²C bus. The LFCSP package has two options for I²C addressing.

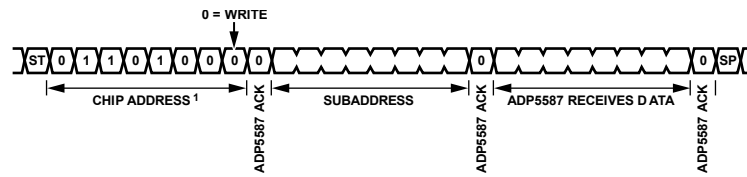
The default part option I²C write address is located at 0x68 (0b01101000), and the read address is at 0x69 (0b 01101001).

The ADP5587ACPZ-1-R7 has the I²C write address at 0x60 (0b 01100000) and the read address at 0x61 (0b 01100001).

All communication to the ADP5587 is performed via its I²C-compatible serial interface. Figure 16 shows a typical write sequence for programming an internal register. The cycle begins with a start condition followed by the chip write address. The ADP5587 acknowledges the chip write address byte by pulling the data line low. The address of the register to which data is to be written is sent next. The ADP5587 acknowledges the register address byte by pulling the data line low. The data

byte to be written is sent next. The ADP5587 acknowledges the data byte by pulling the data line low, and a stop condition completes the sequence.

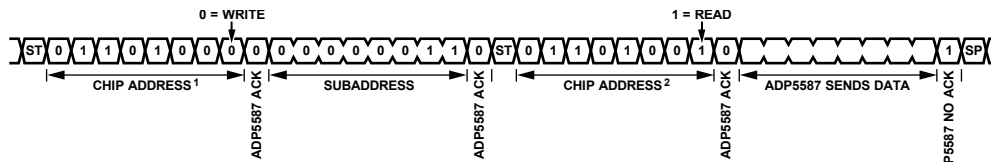
Figure 17 shows a typical read sequence for reading back an internal register. The cycle begins with a start condition followed by the chip write address. The ADP5587 acknowledges the chip write address byte by pulling the data line low. The address of the register from which data is to be read is sent next. The ADP5587 acknowledges the register address byte by pulling the data line low. The cycle continues with a repeat start followed by the chip read address. The ADP5587 acknowledges the chip read address byte by pulling the data line low. The ADP5587 places the contents of the previously addressed register on the bus for readback. There is no acknowledge following the readback data byte, and a stop condition completes the cycle.



¹DEFAULT WRITE ADDRESS IS 0x68. OPTION 1 WRITE ADDRESS IS 0x60.

08612-012

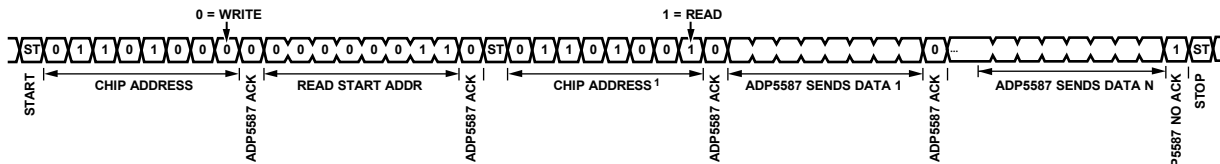
Figure 16. I²C Write Sequence



¹DEFAULT WRITE ADDRESS IS 0x68. OPTION 1 WRITE ADDRESS IS 0x60.
²DEFAULT READ ADDRESS IS 0x69. OPTION 1 READ ADDRESS IS 0x61.

08612-013

Figure 17. I²C Read and Write Sequences



¹DEFAULT READ ADDRESS IS 0x69. OPTION 1 READ ADDRESS IS 0x61.

08612-014

Figure 18. I²C Read Auto-Increment

REGISTERS

The general behavior of registers is as follows:

- All registers are 0 on reset.
- All registers are read/write unless otherwise specified.
- Unused bits are read as 0.
- Interrupt bits are cleared by writing 1 to the flag; writing 0 or reading the flag has no effect, with the exception of the key press, key release, and GPIO interrupt status registers, which are cleared on a read.

Table 15.

| Address | Register Name | Description |
|---------|-----------------|--|
| 0x00 | DEV_ID | Device ID |
| 0x01 | CFG | Configuration Register 1 |
| 0x02 | INT_STAT | Interrupt status register |
| 0x03 | KEY_LCK_EC_STAT | Keylock and event counter register |
| 0x04 | KEY_EVENTA | Key Event Register A |
| 0x05 | KEY_EVENTB | Key Event Register B |
| 0x06 | KEY_EVENTC | Key Event Register C |
| 0x07 | KEY_EVENTD | Key Event Register D |
| 0x08 | KEY_EVENTE | Key Event Register E |
| 0x09 | KEY_EVENTF | Key Event Register F |
| 0x0A | KEY_EVENTG | Key Event Register G |
| 0x0B | KEY_EVENTH | Key Event Register H |
| 0x0C | KEY_EVENTI | Key Event Register I |
| 0x0D | KEY_EVENTJ | Key Event Register J |
| 0x0E | KP_LCK_TMR | Keypad Unlock 1 timer to Keypad Unlock 2 timer |
| 0x0F | UNLOCK1 | Unlock Key 1 |
| 0x10 | UNLOCK2 | Unlock Key 2 |
| 0x11 | GPIO_INT_STAT1 | GPIO interrupt status |
| 0x12 | GPIO_INT_STAT2 | GPIO interrupt status |
| 0x13 | GPIO_INT_STAT3 | GPIO interrupt status |
| 0x14 | GPIO_DAT_STAT1 | GPIO data status, read twice to clear |
| 0x15 | GPIO_DAT_STAT2 | GPIO data status, read twice to clear |
| 0x16 | GPIO_DAT_STAT3 | GPIO data status, read twice to clear |
| 0x17 | GPIO_DAT_OUT1 | GPIO data out |
| 0x18 | GPIO_DAT_OUT2 | GPIO data out |
| 0x19 | GPIO_DAT_OUT3 | GPIO data out |
| 0x1A | GPIO_INT_EN1 | GPIO interrupt enable |
| 0x1B | GPIO_INT_EN2 | GPIO interrupt enable |
| 0x1C | GPIO_INT_EN3 | GPIO interrupt enable |
| 0x1D | KP_GPIO1 | Keypad or GPIO selection |
| 0x1E | KP_GPIO2 | Keypad or GPIO selection |
| 0x1F | KP_GPIO3 | Keypad or GPIO selection |
| 0x20 | GPI_EM_REG1 | GPI Event Mode 1 |
| 0x21 | GPI_EM_REG2 | GPI Event Mode 2 |
| 0x22 | GPI_EM_REG3 | GPI Event Mode 3 |
| 0x23 | GPIO_DIR1 | GPIO data direction |
| 0x24 | GPIO_DIR2 | GPIO data direction |
| 0x25 | GPIO_DIR3 | GPIO data direction |
| 0x26 | GPIO_INT_LVL1 | GPIO level detect |
| 0x27 | GPIO_INT_LVL2 | GPIO level detect |
| 0x28 | GPIO_INT_LVL3 | GPIO level detect |
| 0x29 | DEBOUNCE_DIS1 | Debounce disable |

| Address | Register Name | Description |
|---------|---------------|-------------------|
| 0x2A | DEBOUNCE_DIS2 | Debounce disable |
| 0x2B | DEBOUNCE_DIS3 | Debounce disable |
| 0x2C | GPIO_PULL1 | GPIO pull disable |
| 0x2D | GPIO_PULL2 | GPIO pull disable |
| 0x2E | GPIO_PULL3 | GPIO pull disable |

REGISTER DESCRIPTIONS

Table 16. DEV_ID—Register 0x00 (Device ID)

Default: 0000 XXXX (where X = don't care).

| Register Name | Register Description | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---------------|-----------------------------|-------|-------|-------|-------|-------|-------|-------|-------|
| DEV_ID | Device ID[3:0], MFG ID[7:4] | MFID3 | MFID2 | MFID1 | MFID0 | DID3 | DID2 | DID1 | DID0 |

Table 17. CFG—Register 0x01 (Configuration Register 1)

| Field | Bits | Description |
|--------------|------|---|
| AUTO_INC | 7 | I ² C auto-increment. Burst read is supported; burst write is not supported. 1: I ² C auto-increment is on. 0: I ² C auto-increment is off. |
| GPIEM_CFG | 6 | GPI event mode configuration. 1: GPI events are not tracked when the keypad is locked. 0: GPI events are tracked when the keypad is locked. |
| OVR_FLOW_M | 5 | Overflow mode. 1: overflow mode is on; register overflow data shifts in, starting at the last event and losing first event data. 0: overflow mode is off; register overflow data is lost. |
| INT_CFG | 4 | Interrupt configuration. 1: processor interrupt is deasserted for 275 μ s and is reasserted with pending key events. 0: processor interrupt remains asserted when host tries to clear interrupt while there is a pending key event. |
| OVR_FLOW_IEN | 3 | Overflow interrupt enable. 1: overflow interrupt is enabled. 0: overflow interrupt is disabled. |
| K_LCK_IM | 2 | Keypad lock interrupt mask. 1: keypad lock interrupt is enabled. 0: keypad lock interrupt is disabled. |
| GPI_IEN | 1 | GPI interrupt enable. 1: GPI interrupt is enabled. 0: GPI interrupt is disabled. |
| KE_IEN | 0 | Key events interrupt enable. 1: key events interrupt is enabled. 0: key events interrupt is disabled. |

Table 18. INT_STAT—Register 0x02 (Interrupt Status Register)

| Field | Bits | Description |
|---------------------------|-------|--|
| Not Used | [7:4] | N/A |
| OVR_FLOW_INT ¹ | 3 | Overflow interrupt status. When set, write 1 to clear. 1: overflow interrupt is detected. 0: overflow interrupt is not detected. |
| K_LCK_INT ² | 2 | Keylock interrupt status. When set, write 1 to clear. 1: keylock interrupt is detected. 0: keylock interrupt is not detected. |
| GPI_INT ^{1, 3} | 1 | GPI interrupt status. When set, write 1 to clear. 1: GPI interrupt is detected. 0: GPI interrupt is not detected. |
| KE_INT ^{1, 3} | 0 | Key events interrupt status. When set, write 1 to clear. 1: key events interrupt is detected. 0: key events interrupt is not detected. |

¹ The KE_INT, GPI_INT, and OVR_FLOW_INT bits reflect the status of the interrupts when the interrupt types are enabled even if the processor interrupt is masked.

² The K_LCK_INT bit is the interrupt to the processor when the keypad lock sequence is triggered.

³ If there is a pending key event or GPI interrupt in their respective registers, KE_INT is not cleared until the FIFO is empty, and GPI_INT is not cleared until the cause of the interrupt is resolved. The host must write a 1 to the KE_INT and GPI_INT bits to clear them.

Table 19. KEY_LCK_EC_STAT—Register 0x03 (Keylock and Event Counter Register)

| Field | Bits | Description |
|------------------|-------|---|
| K_LCK_EN | [6] | 0: lock feature is disabled. 1: lock feature is enabled. |
| LCK2, LCK1 | [5:4] | Keypad lock status[1:0] (00 = unlocked; 11 = locked; read-only bits). |
| KEC ¹ | [3:0] | Key event count of key event register. |

¹ The KEC field indicates the key event count of key event registers that have values in the bit (KEC(0000) = 0 events, KEC(0001) = 1 event, KEC(1010) = 10 events). As the key events are read and cleared, the state machine automatically reduces the event count in KEC.

Table 20. KEY_EVENTx—Register 0x04 to Register 0x0D (Key Event Register A to Key Event Register J)¹

| Register Name | Register Description | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|--|--|-------|-------|-------|-------|-------|-------|-------|-------|
| KEY_EVENTA (Register 0x04) | Key Event Register A status (KE[6:0] = key number), KP[7] = 0: released, 1: pressed (cleared on read) | KA7 | KA6 | KA5 | KA4 | KA3 | KA2 | KA1 | KA0 |
| KEY_EVENTB (Register 0x05) | Key Event Register B status (KE[6:0] = key number), KP[7] = 0: released, 1: pressed (cleared on read) | KB7 | KB6 | KB5 | KB4 | KB3 | KB2 | KB1 | KB0 |
| KEY_EVENTC (Register 0x06) | Key Event Register C status (KE[6:0] = key number), KP[7] = 0: released, 1: pressed (cleared on read) | KC7 | KC6 | KC5 | KC4 | KC3 | KC2 | KC1 | KC0 |
| KEY_EVENTD (Register 0x07) | Key Event Register D status (KE[6:0] = key number), KP[7] = 0: released, 1: pressed (cleared on read) | KD7 | KD6 | KD5 | KD4 | KD3 | KD2 | KD1 | KD0 |
| KEY_EVENTE ² (Register 0x08) | Key Event Register E status (KE[6:0] = key number), KP[7] = 0: released, 1: pressed (cleared on read) | KE7 | KE6 | KE5 | KE4 | KE3 | KE2 | KE1 | KE0 |
| KEY_EVENTF (Register 0x09) | Key Event Register F status (KE[6:0] = key number), KP[7] = 0: released, 1: pressed (cleared on read) | KF7 | KF6 | KF5 | KF4 | KF3 | KF2 | KF1 | KF0 |
| KEY_EVENTG (Register 0x0A) | Key Event Register G status (KE[6:0] = key number), KP[7] = 0: released, 1: pressed (cleared on read) | KG7 | KG6 | KG5 | KG4 | KG3 | KG2 | KG1 | KG0 |
| KEY_EVENTH (Register 0x0B) | Key Event Register H status (KE[6:0] = key number), KP[7] = 0: released, 1: pressed (cleared on read) | KH7 | KH6 | KH5 | KH4 | KH3 | KH2 | KH1 | KH0 |
| KEY_EVENTI (Register 0x0C) | Key Event Register I status (KE[6:0] = key number), KP[7] = 0: released, 1: pressed (cleared on read) | KI7 | KI6 | KI5 | KI4 | KI3 | KI2 | KI1 | KI0 |
| KEY_EVENTJ (Register 0x0D) | Key Event Register J status (KE[6:0] = key number), KP[7] = 0: released, 1: pressed (cleared on read) | KJ7 | KJ6 | KJ5 | KJ4 | KJ3 | KJ2 | KJ1 | KJ0 |

¹ Data in key event registers is provided as a FIFO, where data is sequentially provided on each read, regardless of an event register read. The user can read the KEY_EVENTA register only for an event count or can read registers sequentially.

² KE[6:0] reflects the value 1 to 80 for key press events and the value 97 to 114 for GPI events. For KE[7:0], 0 = key released event, 1 = key pressed event. For GPIEM_CFG, 0 reflects a change in the GPI from GPI_INT_LVL = true to GPI_INT_LVL = false; 1 reflects a change in the GPI in which the GPI_INT_LVL condition becomes true.

Table 21. KP_LCK_TMR—Register 0x0E (Keypad Unlock 1 Timer to Keypad Unlock 2 Timer)

| Register Name | Register Description | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---------------|---|-------|-------|-------|-------|-------|-------|-------|-------|
| KP_LCK_TMR | Keypad Unlock 1 timer to Keypad Unlock 2 timer[2:0] (0: disabled, 1 sec to 7 sec) Keypad Lock Interrupt Mask Timer[7:3] (0: disabled, 0 sec to 31 sec) ^{1, 2} | KIMT7 | KIMT6 | KIMT5 | KIMT4 | KIMT3 | KLLT2 | KLLT1 | KLLT0 |

¹ When the keypad lock interrupt mask timer is enabled, the user must press two specific keys before a keylock interrupt is generated or keypad events are recorded. After the keypad is locked, the first time that the user presses any key, a key event interrupt is generated. No additional interrupt is generated unless both unlock key sequences are correct; then a keylock interrupt is generated. When the interrupt mask timer is disabled (0), an interrupt is generated only when the correct full unlock sequence is completed.

² The Unlock 1 timer and Unlock 2 timer keys can be either a key sequence or GPIEM_CFG sequence. The unlock timer keys can be programmed with any value of the keys in the keypad matrix or any GPI values that are part of the key event table. The keylock enable bit (Bit 6, Register 0x03) must be set to lock the keypad.

Table 22. UNLOCK1—Register 0x0F (Unlock Key 1)

| Register Name | Register Description | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---------------|---|-------|-------|-------|-------|-------|-------|-------|-------|
| UNLOCK1 | Unlock Key 1[6:0] (contains key number for Unlock Key 1; 0: disabled) | N/A | ULK6 | ULK5 | ULK4 | ULK3 | ULK2 | ULK1 | ULK0 |

Table 23. UNLOCK2—Register 0x10 (Unlock Key 2)

| Register Name | Register Description | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---------------|---|-------|-------|-------|-------|-------|-------|-------|-------|
| UNLOCK2 | Unlock Key 2[6:0] (contains key number for Unlock Key 2; 0: disabled) | N/A | ULK6 | ULK5 | ULK4 | ULK3 | ULK2 | ULK1 | ULK0 |

Table 24. GPIO_INT_STATx—Register 0x11 to Register 0x13 (GPIO Interrupt Status)

| Register Name | Register Description | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|--------------------------------|--|-------|-------|-------|-------|-------|-------|-------|-------|
| GPIO_INT_STAT1 (Register 0x11) | GPIO interrupt status (used to check GPIO interrupt status, cleared on read) | R7IS | R6IS | R5IS | R4IS | R3IS | R2IS | R1IS | R0IS |
| GPIO_INT_STAT2 (Register 0x12) | GPIO interrupt status (used to check GPIO interrupt status, cleared on read) | C7IS | C6IS | C5IS | C4IS | C3IS | C2IS | C1IS | C0IS |
| GPIO_INT_STAT3 (Register 0x13) | GPIO interrupt status (used to check GPIO interrupt status, cleared on read) | N/A | N/A | N/A | N/A | N/A | N/A | C9IS | C8IS |

Table 25. GPIO_DAT_STATx—Register 0x14 to Register 0x16 (GPIO Data Status)

| Register Name | Register Description | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|--------------------------------|--|-------|-------|-------|-------|-------|-------|-------|-------|
| GPIO_DAT_STAT1 (Register 0x14) | GPIO data status (shows GPIO state when read for inputs) | R7DS | R6DS | R5DS | R4DS | R3DS | R2DS | R1DS | R0DS |
| GPIO_DAT_STAT2 (Register 0x15) | GPIO data status (shows GPIO state when read for inputs) | C7DS | C6DS | C5DS | C4DS | C3DS | C2DS | C1DS | C0DS |
| GPIO_DAT_STAT3 (Register 0x16) | GPIO data status (shows GPIO state when read for inputs) | N/A | N/A | N/A | N/A | N/A | N/A | C9DS | C8DS |

Table 26. GPIO_DAT_OUTx—Register 0x17 to Register 0x19 (GPIO Data Out)

| Register Name | Register Description | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|-------------------------------|---|-------|-------|-------|-------|-------|-------|-------|-------|
| GPIO_DAT_OUT1 (Register 0x17) | GPIO data out (GPIO data to be written to GPIO out driver, inputs are not affected). This is needed so that the value can be written prior to being set as an output. | R7DO | R6DO | R5DO | R4DO | R3DO | R2DO | R1DO | R0DO |
| GPIO_DAT_OUT2 (Register 0x18) | GPIO data out (GPIO data to be written to GPIO out driver, inputs are not affected). This is needed so that the value can be written prior to being set as an output. | C7DO | C6DO | C5DO | C4DO | C3DO | C2DO | C1DO | C0DO |
| GPIO_DAT_OUT3 (Register 0x19) | GPIO data out (GPIO data to be written to GPIO out driver, inputs are not affected). This is needed so that the value can be written prior to being set as an output. | N/A | N/A | N/A | N/A | N/A | N/A | C9DO | C8DO |

Table 27. GPIO_INT_ENx—Register 0x1A to Register 0x1C (GPIO Interrupt Enable)

| Register Name | Register Description | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---------------------------------|---|-------|-------|-------|-------|-------|-------|-------|-------|
| GPIO_INT_EN1 (Register 0x1A) | GPIO interrupt enable (enables interrupts for GP inputs only) | R7IE | R6IE | R5IE | R4IE | R3IE | R2IE | R1IE | R0IE |
| GPIO_INT_EN2 (Register 0x1B) | GPIO interrupt enable (enables interrupts for GP inputs only) | C7IE | C6IE | C5IE | C4IE | C3IE | C2IE | C1IE | C0IE |
| GPIO_INT_EN3 (Register 0x1C) | GPIO interrupt enable (enables interrupts for GP inputs only) | N/A | N/A | N/A | N/A | N/A | N/A | C9IE | C8IE |

Table 28. KP_GPIOx—Register 0x1D to Register 0x1F (Keypad or GPIO Selection)

| Register Name | Register Description | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|-----------------------------|---|-------|-------|-------|-------|-------|-------|-------|-------|
| KP_GPIO1 (Register 0x1D) | Keypad or GPIO selection 0: GPIO 1: KP matrix | R7 | R6 | R5 | R4 | R3 | R2 | R1 | R0 |
| KP_GPIO2 (Register 0x1E) | Keypad or GPIO selection 0: GPIO 1: KP matrix | C7 | C6 | C5 | C4 | C3 | C2 | C1 | C0 |
| KP_GPIO3 (Register 0x1F) | Keypad or GPIO selection 0: GPIO 1: KP matrix | N/A | N/A | N/A | N/A | N/A | N/A | C9 | C8 |

Table 29. GPI_EM_REGx—Register 0x20 to Register 0x22 (GPI Event Mode 1 to GPI Event Mode 3)

| Register Name | Register Description | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|--------------------------------|--|-------|-------|-------|-------|-------|-------|-------|-------|
| GPI_EM_REG1 (Register 0x20) | GPI Event Mode Register 1 0: GPI not part of event FIFO 1: GPI part of event FIFO (R0 to R7) | R7_EM | R6_EM | R5_EM | R4_EM | R3_EM | R2_EM | R1_EM | R0_EM |
| GPI_EM_REG2 (Register 0x21) | GPI Event Mode Register 2 0: GPI not part of event FIFO 1: GPI part of event FIFO (C0 to C7) | C7_EM | C6_EM | C5_EM | C4_EM | C3_EM | C2_EM | C1_EM | C0_EM |
| GPI_EM_REG3 (Register 0x22) | GPI Event Mode Register 3 0: GPI not part of event FIFO 1: GPI part of event FIFO (C8 to C9) | NA | NA | NA | NA | NA | NA | C9_EM | C8_EM |

Table 30. GPIO_DIRx—Register 0x23 to Register 0x25 (GPIO Data Direction)

| Register Name | Register Description | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|------------------------------|--|-------|-------|-------|-------|-------|-------|-------|-------|
| GPIO_DIR1 (Register 0x23) | GPIO data direction 0: input 1: output | R7D | R6D | R5D | R4D | R3D | R2D | R1D | R0D |
| GPIO_DIR2 (Register 0x24) | GPIO data direction 0: input 1: output | C7D | C6D | C5D | C4D | C3D | C2D | C1D | C0D |
| GPIO_DIR3 (Register 0x25) | GPIO data direction 0: input 1: output | N/A | N/A | N/A | N/A | N/A | N/A | C9D | C8D |

Table 31. GPIO_INT_LVLx—Register 0x26 to Register 0x28 (GPIO Level Detect)

| Register Name | Register Description | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|----------------------------------|--|-------|-------|-------|-------|-------|-------|-------|-------|
| GPIO_INT_LVL1 (Register 0x26) | GPIO INT level detect 0: low 1: high | R7IL | R6IL | R5IL | R4IL | R3IL | R2IL | R1IL | R0IL |
| GPIO_INT_LVL2 (Register 0x27) | GPIO INT level detect 0: low 1: high | C7IL | C6IL | C5IL | C4IL | C3IL | C2IL | C1IL | C0IL |
| GPIO_INT_LVL3 (Register 0x28) | GPIO INT level detect 0: low 1: high | N/A | N/A | N/A | N/A | N/A | N/A | C9IL | C8IL |

Table 32. DEBOUNCE_DISx—Register 0x29 to Register 0x2B (Debounce Disable)

| Register Name | Register Description | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|----------------------------------|--|-------|-------|-------|-------|-------|-------|-------|-------|
| DEBOUNCE_DIS1 (Register 0x29) | Debounce disable (inputs) 0: enabled 1: disabled | R7DD | R6DD | R5DD | R4DD | R3DD | R2DD | R1DD | R0DD |
| DEBOUNCE_DIS2 (Register 0x2A) | Debounce disable (inputs) 0: enabled 1: disabled | C7DD | C6DD | C5DD | C4DD | C3DD | C2DD | C1DD | C0DD |
| DEBOUNCE_DIS3 (Register 0x2B) | Debounce disable (inputs) 0: enabled 1: disabled | N/A | N/A | N/A | N/A | N/A | N/A | C9DD | C8DD |

Table 33. GPIO_PULLx—Register 0x2C to Register 0x2E (GPIO Pull Disable)

| Register Name | Register Description | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|-------------------------------|--|-------|-------|-------|-------|-------|-------|-------|-------|
| GPIO_PULL1 (Register 0x2C) | GPIO pull disable (remove pull-ups from inputs) 0: pull enabled 1: pull disabled | R7PD | R6PD | R5PD | R4PD | R3PD | R2PD | R1PD | R0PD |
| GPIO_PULL2 (Register 0x2D) | GPIO pull disable (remove pull-ups from inputs) 0: pull enabled 1: pull disabled | C7PD | C6PD | C5PD | C4PD | C3PD | C2PD | C1PD | C0PD |
| GPIO_PULL3 (Register 0x2E) | GPIO pull disable (remove pull-ups from inputs) 0: pull enabled 1: pull disabled | N/A | N/A | N/A | N/A | N/A | N/A | C9PD | C8PD |

APPLICATIONS INFORMATION

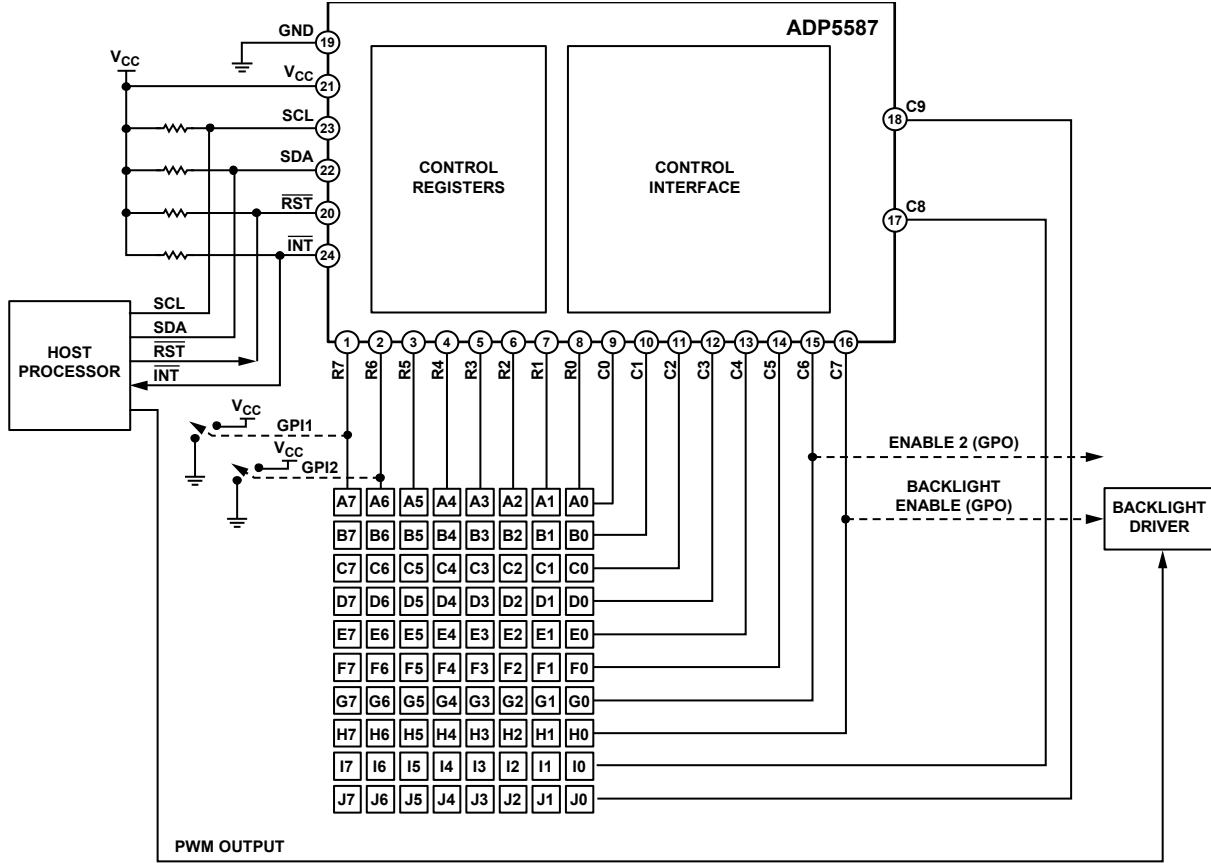


Figure 19. ADP5587 Detailed Application Block Diagram

APPLICATIONS OVERVIEW

The ADP5587 is designed to complement host processors in a variety of ways. Its versatility makes it the ideal solution for mobile platforms that require extended keypads and GPIO expanders. The programmable registers give the designer the flexibility to configure any or all of its GPIOs in a variety of ways. Figure 19 shows a detailed application diagram.

KEYPAD CURRENT

Keypad current drain varies based on how many keys and how many rows and columns are pressed during multiple key presses. Table 34 shows the typical current drain for a single key press and for two key presses.

Table 34. Typical Current Drain

| Number of Key Presses | Conditions ¹ | Typical | Unit |
|-----------------------|----------------------------------|---------|------|
| 1 | V _{CC} = 1.8 V to 3.0 V | 55 | μA |
| 2 | V _{CC} = 1.8 V to 3.0 V | 100 | μA |

¹ T_A = T_J = -40°C to +85°C.

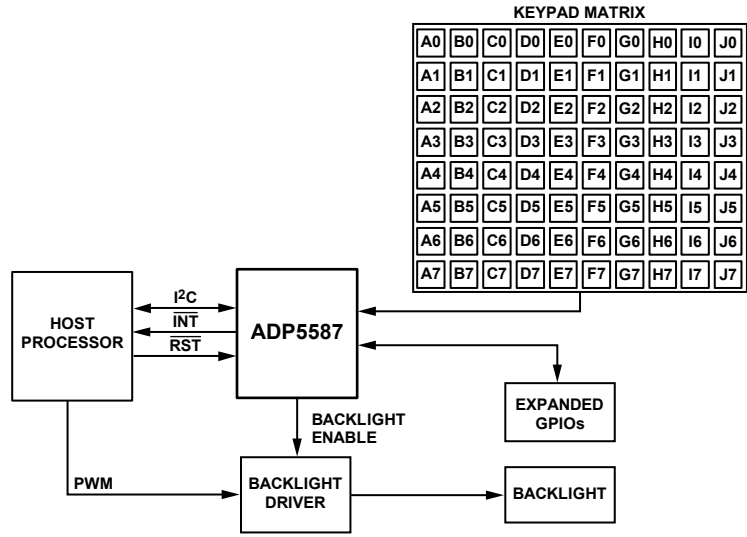
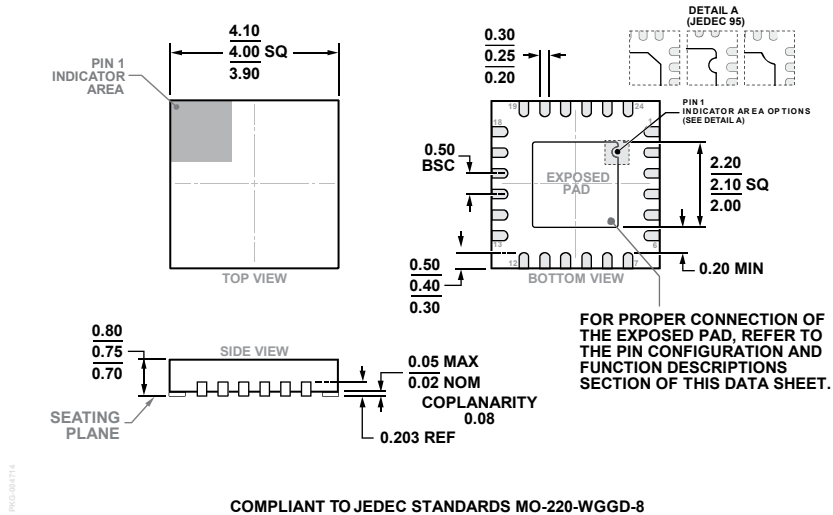


Figure 20. Integration Block Diagram

08612-016

OUTLINE DIMENSIONS



COMPLIANT TO JEDEC STANDARDS MO-220-WGGD-8

Figure 21. 24-Lead Lead Frame Chip Scale Package [LFCS]
4 mm × 4 mm Body and 0.75 mm Package Height
(CP-24-10)
Dimensions shown in millimeters

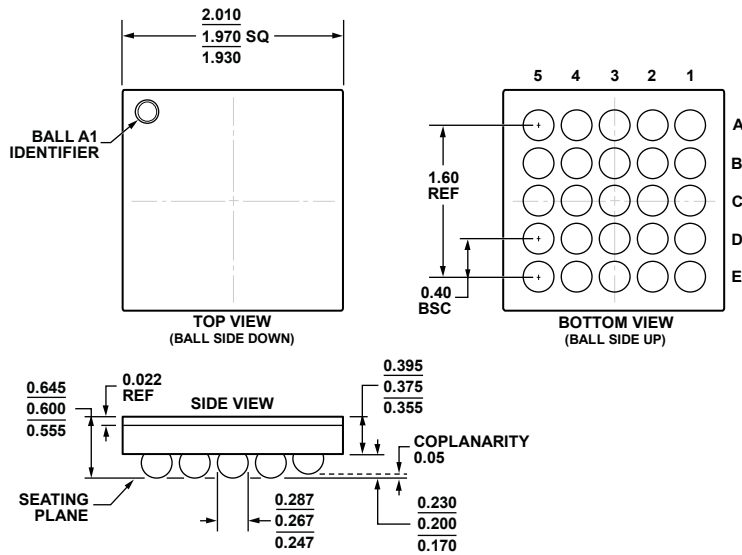


Figure 22. 25-Ball Wafer Level Chip Scale Package [WLCSP]
(CB-25-4)
Dimensions shown in millimeters

ORDERING GUIDE

| Model ¹ | Temperature Range | I ² C Write Address | | Package Description | Package Option |
|--------------------|-------------------|--------------------------------|------|--|----------------|
| | | Write | Read | | |
| ADP5587ACPZ-R7 | -40°C to +85°C | 0x68 | 0x69 | 24-Lead Lead Frame Chip Scale Package [LFCS] | CP-24-10 |
| ADP5587ACPZ-1-R7 | -40°C to +85°C | 0x60 | 0x61 | 24-Lead Lead Frame Chip Scale Package [LFCS] | CP-24-10 |
| ADP5587ACBZ-R7 | -40°C to +85°C | 0x68 | 0x69 | 25-Ball Wafer Level Chip Scale Package [WLCSP] | CB-25-4 |
| ADP5587CP-EVALZ | | 0x68 | 0x69 | Evaluation Board [LFCS] | |
| ADP5587CB-EVALZ | | 0x68 | 0x69 | Evaluation Board [WLCSP] | |

¹ Z = RoHS Compliant Part.

NOTES

I²C refers to a communications protocol originally developed by Philips Semiconductors (now NXP Semiconductors).