

## 具有集成 ESD 保护的 TCA8418 I<sup>2</sup>C 控制型键盘扫描 IC

### 1 特性

- 工作电源电压范围：1.65V 至 3.6V
- 可借助 18 个通用输入输出 (GPIO) 支持 80 个按钮
- 支持 QWERTY 键盘操作和通用输入输出 (GPIO) 扩展
- 低待机（空闲）电流消耗：3 $\mu$ A
- 支持 1MHz 快速模式加上 I<sup>2</sup>C 总线
- 10 字节先入先出 (FIFO) 可存储 10 次按键按压和释放操作
- 开漏电路低电平有效中断输出
- 集成消抖时间为 50 $\mu$ s
- 施密特触发器操作可在 SCL 和 SDA 输入端实现缓慢输入转换和更好的开关噪声抗扰度：1.8V 时的  $V_{hys}$  典型值为 0.18V
- 锁断性能超过 200mA，符合 JESD 78 II 类规范的要求
- 全部 18 个 GPIO 引脚和非 GPIO 引脚上的 ESD 保护均超出 JESD 22 规范要求
  - 2000V 人体放电模型 (A114-A)
  - 1000V 器件充电模型 (C101)

### 2 应用范围

- 智能电话
- 平板电脑
- 人机界面 (HMI) 面板
- 全球卫星定位 (GPS) 设备
- MP3 播放器
- 数码照相机

### 3 说明

TCA8418 是一款具有集成 ESD 保护的键盘扫描器件。它能够在 1.65V 至 3.6V 电源电压下运行，具有 18 个 GPIO。这些 GPIO 可通过 I<sup>2</sup>C 接口支持多达 80 个按键。

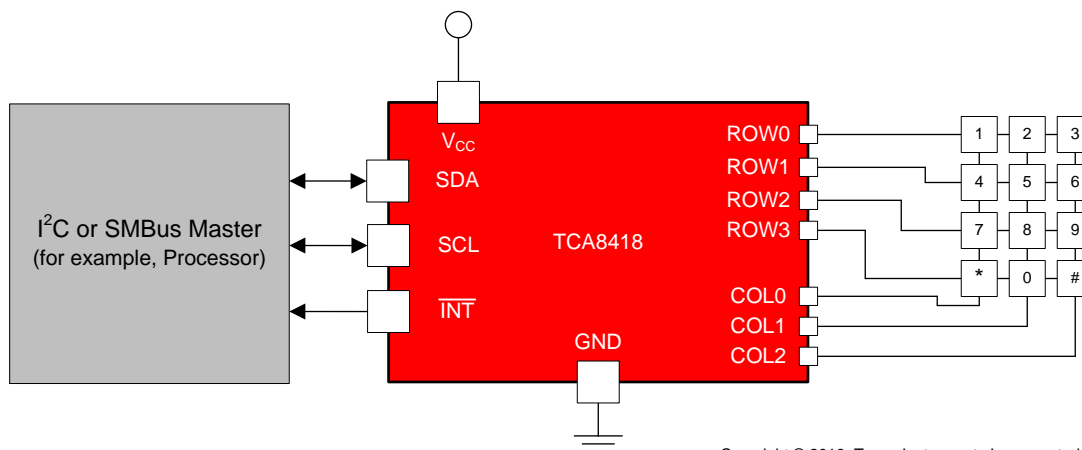
按键控制器可对输入进行消抖以及保持用于存储按键按压和释放事件的 10 字节 FIFO。该 FIFO 借助溢出绕回功能可存储多达 10 个按键事件。中断 ( $\overline{INT}$ ) 输出可配置为在发生按键按压和按键释放事件时，或这些事件达到最大速率时报警。

#### 器件信息<sup>(1)</sup>

| 器件型号    | 封装        | 封装尺寸 (标称值)      |
|---------|-----------|-----------------|
| TCA8418 | WQFN (24) | 4.00mm x 4.00mm |

(1) 如需了解所有可用封装，请参阅产品说明书末尾的可订购产品附录。

简化原理图



仅显示 7 个 GPIO (共 18 个)



## 目录

|          |  |           |           |   |           |
|----------|--|-----------|-----------|---|-----------|
| <b>1</b> | 特性 .....   | <b>1</b>  | 8.2       | Functional Block Diagram .....              | <b>15</b> |
| <b>2</b> | 应用范围 .....   | <b>1</b>  | 8.3       | Feature Description .....                   | <b>15</b> |
| <b>3</b> | 说明 .....   | <b>1</b>  | 8.4       | Device Functional Modes .....               | <b>20</b> |
| <b>4</b> | 修订历史记录 .....   | <b>2</b>  | 8.5       | Programming .....                           | <b>21</b> |
| <b>5</b> | <b>Pin Configuration and Functions</b> .....         | <b>3</b>  | 8.6       | Register Maps .....                         | <b>25</b> |
| <b>6</b> | <b>Specifications</b> .....                          | <b>4</b>  | <b>9</b>  | <b>Application and Implementation</b> ..... | <b>37</b> |
| 6.1      | Absolute Maximum Ratings .....                       | <b>4</b>  | 9.1       | Application Information .....               | <b>37</b> |
| 6.2      | ESD Ratings .....                                    | <b>4</b>  | 9.2       | Typical Application .....                   | <b>39</b> |
| 6.3      | Recommended Operating Conditions .....               | <b>4</b>  | <b>10</b> | <b>Power Supply Recommendations</b> .....   | <b>42</b> |
| 6.4      | Thermal Information .....                            | <b>4</b>  | <b>11</b> | <b>Layout</b> .....                         | <b>44</b> |
| 6.5      | Electrical Characteristics .....                     | <b>5</b>  | 11.1      | Layout Guidelines .....                     | <b>44</b> |
| 6.6      | I <sup>2</sup> C Interface Timing Requirements ..... | <b>6</b>  | 11.2      | Layout Example .....                        | <b>44</b> |
| 6.7      | Reset Timing Requirements .....                      | <b>6</b>  | <b>12</b> | <b>器件和文档支持</b> .....                        | <b>45</b> |
| 6.8      | Switching Characteristics .....                      | <b>7</b>  | 12.1      | 接收文档更新通知 .....                              | <b>45</b> |
| 6.9      | Keypad Switching Characteristics .....               | <b>7</b>  | 12.2      | 社区资源 .....                                  | <b>45</b> |
| 6.10     | Typical Characteristics .....                        | <b>8</b>  | 12.3      | 商标 .....                                    | <b>45</b> |
| <b>7</b> | <b>Parameter Measurement Information</b> .....       | <b>11</b> | 12.4      | 静电放电警告 .....                                | <b>45</b> |
| <b>8</b> | <b>Detailed Description</b> .....                    | <b>15</b> | 12.5      | 术语表 .....                                   | <b>45</b> |
| 8.1      | Overview .....                                       | <b>15</b> | <b>13</b> | <b>机械、封装和可订购信息</b> .....                    | <b>45</b> |

## 4 修订历史记录

注：之前版本的页码可能与当前版本有所不同。

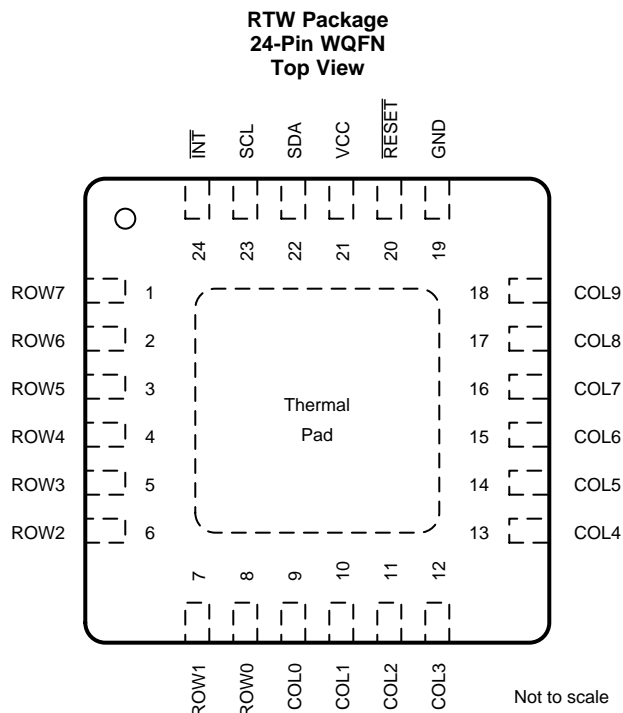
| <b>Changes from Revision F (July 2017) to Revision G</b>   | <b>Page</b> |
|--|-------------|
| • Added sentence: " If debouncing is enabled, these registers return..." to <a href="#">GPIO Data Status Registers, GPIO_DAT_STAT1–3 (Address 0x14–0x16)</a> ..... | <b>31</b>   |

| <b>Changes from Revision E (November 2015) to Revision F</b>                       | <b>Page</b> |
|--|-------------|
| • 将 WQFN 封装尺寸从“4.00mm × 3.00mm”更改为“4.00mm × 4.00mm”，删除了器件信息表中的 DSBGA (25) 封装 ..... | <b>1</b>    |

| <b>Changes from Revision D (July 2014) to Revision E</b>                           | <b>Page</b> |
|--|-------------|
| • 已添加 ESD 额定值表，特性描述部分、器件功能模式、应用和实施部分、电源相关建议部分、布局部分、器件和文档支持部分以及机械、封装和可订购信息部分。 ..... | <b>1</b>    |
| • Updated <i>Register Descriptions</i> table. ....                                 | <b>25</b>   |

| <b>Changes from Revision B (March 2010) to Revision C</b> | <b>Page</b> |
|---|-------------|
| • Added CAD Interrupt Errata section. ....                | <b>35</b>   |
| • Added Overflow Errata section. ....                     | <b>36</b>   |

## 5 Pin Configuration and Functions



### Pin Functions

| PIN |                           | TYPE | DESCRIPTION  |
|-----|---------------------------|------|--|
| No. | NAME                      |      |  |
| 1   | ROW7                      | I/O  | GPIO or row 7 in keypad matrix. If unused, connect to $V_{CC}$ through a pull-up resistor.               |
| 2   | ROW6                      | I/O  | GPIO or row 6 in keypad matrix. If unused, connect to $V_{CC}$ through a pull-up resistor.               |
| 3   | ROW5                      | I/O  | GPIO or row 5 in keypad matrix. If unused, connect to $V_{CC}$ through a pull-up resistor.               |
| 4   | ROW4                      | I/O  | GPIO or row 4 in keypad matrix. If unused, connect to $V_{CC}$ through a pull-up resistor.               |
| 5   | ROW3                      | I/O  | GPIO or row 3 in keypad matrix. If unused, connect to $V_{CC}$ through a pull-up resistor.               |
| 6   | ROW2                      | I/O  | GPIO or row 2 in keypad matrix. If unused, connect to $V_{CC}$ through a pull-up resistor.               |
| 7   | ROW1                      | I/O  | GPIO or row 1 in keypad matrix. If unused, connect to $V_{CC}$ through a pull-up resistor.               |
| 8   | ROW0                      | I/O  | GPIO or row 0 in keypad matrix. If unused, connect to $V_{CC}$ through a pull-up resistor.               |
| 9   | COL0                      | I/O  | GPIO or column 0 in keypad matrix. If unused, connect to $V_{CC}$ through a pull-up resistor.            |
| 10  | COL1                      | I/O  | GPIO or column 1 in keypad matrix. If unused, connect to $V_{CC}$ through a pull-up resistor.            |
| 11  | COL2                      | I/O  | GPIO or column 2 in keypad matrix. If unused, connect to $V_{CC}$ through a pull-up resistor.            |
| 12  | COL3                      | I/O  | GPIO or column 3 in keypad matrix. If unused, connect to $V_{CC}$ through a pull-up resistor.            |
| 13  | COL4                      | I/O  | GPIO or column 4 in keypad matrix. If unused, connect to $V_{CC}$ through a pull-up resistor.            |
| 14  | COL5                      | I/O  | GPIO or column 5 in keypad matrix. If unused, connect to $V_{CC}$ through a pull-up resistor.            |
| 15  | COL6                      | I/O  | GPIO or column 6 in keypad matrix. If unused, connect to $V_{CC}$ through a pull-up resistor.            |
| 16  | COL7                      | I/O  | GPIO or column 7 in keypad matrix. If unused, connect to $V_{CC}$ through a pull-up resistor.            |
| 17  | COL8                      | I/O  | GPIO or column 8 in keypad matrix. If unused, connect to $V_{CC}$ through a pull-up resistor.            |
| 18  | COL9                      | I/O  | GPIO or column 9 in keypad matrix. If unused, connect to $V_{CC}$ through a pull-up resistor.            |
| 19  | GND                       | –    | Ground   |
| 20  | $\overline{\text{RESET}}$ | I    | Active-low reset input. Connect to $V_{CC}$ through a pull-up resistor, if no active connection is used. |
| 21  | $V_{CC}$                  | -    | Supply voltage of 1.65 V to 3.6 V  |
| 22  | SDA                       | I/O  | Serial data bus. Connect to $V_{CC}$ through a pull-up resistor.   |
| 23  | SCL                       | I    | Serial clock bus. Connect to $V_{CC}$ through a pull-up resistor.  |
| 24  | $\overline{\text{INT}}$   | O    | Active-low interrupt output. Open drain structure. Connect to $V_{CC}$ through a pull-up resistor.       |

## 6 Specifications

### 6.1 Absolute Maximum Ratings<sup>(1)</sup>

over operating free-air temperature range (unless otherwise noted)

|                  |   | MIN                     | MAX                                   | UNIT   |
|------------------|---|-------------------------|---------------------------------------|--------|
| V <sub>CC</sub>  | Supply voltage range  | -0.5                    | 4.6                                   | V      |
| V <sub>I</sub>   | Input voltage range <sup>(2)</sup>  | -0.5                    | 4.6                                   | V      |
| V <sub>O</sub>   | Voltage range applied to any output in the high-impedance or power-off state <sup>(2)</sup> | -0.5                    | 4.6                                   | V      |
|                  | Output voltage range in the high or low state <sup>(2)</sup>                                | -0.5                    | 4.6                                   |        |
| I <sub>IK</sub>  | Input clamp current   | V <sub>I</sub> < 0      |                                       | ±20 mA |
| I <sub>OK</sub>  | Output clamp current  | V <sub>O</sub> < 0      |                                       | ±20 mA |
| I <sub>OL</sub>  | Continuous output Low current   | P port, SDA             | V <sub>O</sub> = 0 to V <sub>CC</sub> | 50     |
|                  |   | $\overline{\text{INT}}$ |                                       | 25     |
| I <sub>OH</sub>  | Continuous output High current  | P port                  | V <sub>O</sub> = 0 to V <sub>CC</sub> | 50 mA  |
| T <sub>stg</sub> | Storage temperature range   | -65                     | 150                                   | °C     |

- (1) Stresses beyond those listed under *Absolute Maximum Ratings* may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those indicated under *Recommended Operating Conditions* is not implied. Exposure to absolute-maximum-rated conditions for extended periods may affect device reliability.
- (2) The input negative-voltage and output voltage ratings may be exceeded if the input and output current ratings are observed.

### 6.2 ESD Ratings

|                    |                         | VALUE  | UNIT  |
|--------------------|-------------------------|--|-------|
| V <sub>(ESD)</sub> | Electrostatic discharge | Human body model (HBM), per ANSI/ESDA/JEDEC JS-001, all pins <sup>(1)</sup>              | ±2000 |
|                    |                         | Charged device model (CDM), per JEDEC specification JESD22-C101, all pins <sup>(2)</sup> | ±1000 |

- (1) JEDEC document JEP155 states that 500-V HBM allows safe manufacturing with a standard ESD control process.
- (2) JEDEC document JEP157 states that 250-V CDM allows safe manufacturing with a standard ESD control process.

### 6.3 Recommended Operating Conditions

|                 |                                | MIN   | MAX | UNIT                            |
|-----------------|--------------------------------|---|-----|---------------------------------|
| V <sub>CC</sub> | Supply voltage                 | 1.65  | 3.6 | V                               |
| V <sub>IH</sub> | High-level input voltage       | SCL, SDA, ROW0–7, COL0–9, $\overline{\text{RESET}}$ |     | 0.7 × V <sub>CC</sub> V         |
| V <sub>IL</sub> | Low-level input voltage        | SCL, SDA, ROW0–7, COL0–9, $\overline{\text{RESET}}$ |     | -0.5 to 0.3 × V <sub>CC</sub> V |
| I <sub>OH</sub> | High-level output current      | ROW0–7, COL0–9                                      |     | 10 mA                           |
| I <sub>OL</sub> | Low-level output current       | ROW0–7, COL0–9                                      |     | 25 mA                           |
| T <sub>A</sub>  | Operating free-air temperature | -40   | 85  | °C                              |

### 6.4 Thermal Information

| THERMAL METRIC <sup>(1)</sup> |  | TCA8418    | UNIT |
|-------------------------------|--|------------|------|
|                               |  | RTW (WQFN) |      |
|                               |  | 24 PINS    |      |
| R <sub>θJA</sub>              | Junction-to-ambient thermal resistance       | 38.9       | °C/W |
| R <sub>θJC(top)</sub>         | Junction-to-case (top) thermal resistance    | 40.6       | °C/W |
| R <sub>θJB</sub>              | Junction-to-board thermal resistance         | 17.2       | °C/W |
| ψ <sub>JT</sub>               | Junction-to-top characterization parameter   | 0.9        | °C/W |
| ψ <sub>JB</sub>               | Junction-to-board characterization parameter | 17.2       | °C/W |
| R <sub>θJC(bot)</sub>         | Junction-to-case (bottom) thermal resistance | 6.6        | °C/W |

- (1) For more information about traditional and new thermal metrics, see the [Semiconductor and IC Package Thermal Metrics](#) application report.

## 6.5 Electrical Characteristics

over recommended operating free-air temperature range,  $V_{CC} = 1.65\text{ V to }3.6\text{ V}$  (unless otherwise noted)

| PARAMETER                |  | TEST CONDITIONS   | $V_{CC}$                   | MIN                                     | TYP             | MAX  | UNIT          |               |
|--------------------------|--|---|----------------------------|---|-----------------|------|---------------|---------------|
| $V_{IK}$                 | Input diode clamp voltage                  | $I_I = -18\text{ mA}$   | 1.65 V to 3.6 V            | -1.2                                    |                 |      | V             |               |
| $V_{PORR}$               | Power-on reset voltage, $V_{CC}$ rising    | $V_I = V_{CC}$ or GND, $I_O = 0$  | 1.65 V to 3.6 V            | 1.03                                    |                 | 1.43 | V             |               |
| $V_{PORF}$               | Power-on reset voltage, $V_{CC}$ falling   |   |                            | 0.76                                    | 1.15            |      |               |               |
| $V_{OH}$                 | ROW0–7, COL0–9 high-level output voltage   | $I_{OH} = -1\text{ mA}$   | 1.65 V                     | 1.25                                    |                 |      | V             |               |
|                          |  |   | 1.65 V                     | 1.2                                     |                 |      |               |               |
|                          |  | $I_{OH} = -8\text{ mA}$   | 2.3 V                      | 1.8                                     |                 |      |               |               |
|                          |  |   | 3 V                        | 2.6                                     |                 |      |               |               |
|                          |  | $I_{OH} = -10\text{ mA}$  | 1.65 V                     | 1.1                                     |                 |      |               |               |
|                          |  |   | 2.3 V                      | 1.7                                     |                 |      |               |               |
| $V_{OL}$                 | ROW0–7, COL0–9 low-level output voltage    | $I_{OL} = 1\text{ mA}$  | 1.65 V                     |   |                 | 0.4  | V             |               |
|                          |  |   | 1.65 V                     |   |                 | 0.45 |               |               |
|                          |  | $I_{OL} = 8\text{ mA}$  | 2.3 V                      |   |                 | 0.25 |               |               |
|                          |  |   | 3 V                        |   |                 | 0.25 |               |               |
|                          |  | $I_{OL} = 10\text{ mA}$   | 1.65 V                     |   |                 | 0.6  |               |               |
|                          |  |   | 2.3 V                      |   |                 | 0.3  |               |               |
| $I_{OL}$                 | SDA  | $V_{OL} = 0.4\text{ V}$   | 1.65 V to 3.6 V            | 3                                       |                 |      | mA            |               |
|                          | $\overline{INT}$ and $\overline{CAD\_INT}$ | $V_{OL} = 0.4\text{ V}$   | 1.65 V to 3.6 V            | 3                                       |                 |      |               |               |
| $I_I$                    | SCL, SDA, ROW0–7, COL0–9, RESET            | $V_I = V_{CCI}$ or GND  | 1.65 V to 3.6 V            |   |                 | 1    | $\mu\text{A}$ |               |
| $R_{INT}$                | Internal pull-up resistor value            | ROW0–7, COL0–9  |                            |   | 105             |      | k $\Omega$    |               |
| $I_{CC}$                 | Supply current                             | $V_I$ on SDA, ROW0–7, COL0–9 = $V_{CC}$ or GND, $I_O = 0$ , I/O = inputs, | $f_{SCL} = 0\text{ kHz}$   | Oscillator OFF                          | 1.65 V to 3.6 V |      | 10            | $\mu\text{A}$ |
|                          |  |   |                            | Oscillator ON                           |                 |      | 18            |               |
|                          |  |   | $f_{SCL} = 400\text{ kHz}$ | 1 key press                             | 1.65 V          | 50   |               |               |
|                          |  |   |                            |   | 3.6 V           | 90   |               |               |
|                          |  |   | $f_{SCL} = 1\text{ MHz}$   | 1 key press                             | 1.65 V          | 65   |               |               |
|                          |  |   |                            |   | 3.6 V           | 153  |               |               |
|                          |  |   | $f_{SCL} = 400\text{ kHz}$ | GPI low (pull-up enable) <sup>(1)</sup> | 1.65 V to 3.6 V | 55   |               |               |
|                          |  |   |                            |   |                 | 65   |               |               |
|                          |  |   | $f_{SCL} = 400\text{ kHz}$ | GPI low (pull-up disable)               | 1.65 V to 3.6 V | 15   |               |               |
|                          |  |   |                            |   |                 | 24   |               |               |
| $f_{SCL} = 1\text{ MHz}$ | 1 GPO active                               | 1.65 V to 3.6 V   | 55                         |   |                 |      |               |               |
|                          |  |   | 65                         |   |                 |      |               |               |
| $C_I$                    | SCL  | $V_I = V_{CCI}$ or GND  | 1.65 V to 3.6 V            |   | 6               | 8    | pF            |               |
| $C_{IO}$                 | SDA  | $V_{IO} = V_{CC}$ or GND  | 1.65 V to 3.6 V            |   | 10              | 12.5 | pF            |               |
|                          | ROW0–7, COL0–9                             |   |                            |   | 5               | 6    |               |               |

(1) Assumes that one GPIO is enabled.

## 6.6 I<sup>2</sup>C Interface Timing Requirements

over recommended operating free-air temperature range (unless otherwise noted) (see Figure 16)

|  | STANDARD MODE<br>I <sup>2</sup> C BUS |      | FAST MODE<br>I <sup>2</sup> C BUS |     | FAST MODE PLUS (FM+)<br>I <sup>2</sup> C BUS |      | UNIT |
|--|---------------------------------------|------|-----------------------------------|-----|--|------|------|
|  | MIN                                   | MAX  | MIN                               | MAX | MIN  | MAX  |      |
| $f_{scl}$ I <sup>2</sup> C clock frequency   | 0                                     | 100  | 0                                 | 400 | 0  | 1000 | kHz  |
| $t_{sch}$ I <sup>2</sup> C clock high time   | 4                                     |      | 0.6                               |     | 0.26   |      | μs   |
| $t_{scl}$ I <sup>2</sup> C clock low time  | 4.7                                   |      | 1.3                               |     | 0.5  |      | μs   |
| $t_{sp}$ I <sup>2</sup> C spike time   |                                       | 50   |                                   | 50  |  | 50   | ns   |
| $t_{sds}$ I <sup>2</sup> C serial data setup time  | 250                                   |      | 100                               |     | 50   |      | ns   |
| $t_{sdh}$ I <sup>2</sup> C serial data hold time   | 0                                     |      | 0                                 |     | 0  |      | ns   |
| $t_{icr}$ I <sup>2</sup> C input rise time   |                                       | 1000 | $20 + 0.1C_b^{(1)}$               | 300 |  | 120  | ns   |
| $t_{icf}$ I <sup>2</sup> C input fall time   |                                       | 300  | $20 + 0.1C_b^{(1)}$               | 300 |  | 120  | ns   |
| $t_{ocf}$ I <sup>2</sup> C output fall time; 10 pF to 400 pF bus                         |                                       | 300  | $20 + 0.1C_b^{(1)}$               | 300 |  | 120  | μs   |
| $t_{buf}$ I <sup>2</sup> C bus free time between Stop and Start                          | 4.7                                   |      | 1.3                               |     | 0.5  |      | μs   |
| $t_{sts}$ I <sup>2</sup> C Start or repeater Start condition setup time                  | 4.7                                   |      | 0.6                               |     | 0.26   |      | μs   |
| $t_{sth}$ I <sup>2</sup> C Start or repeater Start condition hold time                   | 4                                     |      | 0.6                               |     | 0.26   |      | μs   |
| $t_{sps}$ I <sup>2</sup> C Stop condition setup time                                     | 4                                     |      | 0.6                               |     | 0.26   |      | μs   |
| $t_{vd(data)}$ Valid data time; SCL low to SDA output valid                              |                                       | 1    |                                   | 0.9 |  | 0.45 | μs   |
| $t_{vd(ack)}$ Valid data time of ACK condition; ACK signal from SCL low to SDA (out) low |                                       | 1    |                                   | 0.9 |  | 0.45 | μs   |

 (1)  $C_b$  = total capacitance of one bus line in pF

## 6.7 Reset Timing Requirements

over recommended operating free-air temperature range (unless otherwise noted) (see Figure 19)

|                               | STANDARD MODE, FAST<br>MODE, FAST MODE PLUS<br>(FM+)<br>I <sup>2</sup> C BUS |     | UNIT |
|-------------------------------|--|-----|------|
|                               | MIN  | MAX |      |
| $t_W$ Reset pulse duration    | $120^{(1)}$  |     | μs   |
| $t_{REC}$ Reset recovery time | $120^{(1)}$  |     | μs   |
| $t_{RESET}$ Time to reset     | $120^{(1)}$  |     | μs   |

(1) The GPIO debounce circuit uses each GPIO input which passes through a two-stage register circuit. Both registers are clocked by the same clock signal, presumably free-running, with a nominal period of 50 μs. When an input changes state, the new state is clocked into the first stage on one clock transition. On the next same-direction transition, if the input state is still the same as the previously clocked state, the signal is clocked into the second stage, and then on to the remaining circuits. Since the inputs are asynchronous to the clock, it will take anywhere from zero to 50 μs after the input transition to clock the signal into the first stage. Therefore, the total debounce time may be as long as 100 μs. Finally, to account for a slow clock, the spec further guard-banded at 120 μs.

## 6.8 Switching Characteristics

| PARAMETER       |                            | FROM           | TO                      | STANDARD MODE, FAST MODE, FAST MODE PLUS (FM+) I <sup>2</sup> C BUS |     | UNIT |
|-----------------|----------------------------|----------------|-------------------------|---|-----|------|
|                 |                            |                |                         | MIN   | MAX |      |
| t <sub>IV</sub> | Interrupt valid time       | ROW0–7, COL0–9 | $\overline{\text{INT}}$ | 20  | 60  | μs   |
|                 |                            |                |                         | 40  | 120 |      |
|                 |                            |                |                         | 10  | 30  |      |
| t <sub>IR</sub> | Interrupt reset delay time | SCL            | $\overline{\text{INT}}$ |   | 200 | ns   |
| t <sub>PV</sub> | Output data valid          | SCL            | ROW0–7, COL0–9          |   | 400 | ns   |
| t <sub>PS</sub> | Input data setup time      | P port         | SCL                     | 0   |     | ns   |
| t <sub>PH</sub> | Input data hold time       | P port         | SCL                     | 300   |     | ns   |

## 6.9 Keypad Switching Characteristics

| PARAMETER                      | STANDARD MODE, FAST MODE, FAST MODE PLUS (FM+) I <sup>2</sup> C BUS |     | UNIT |
|--------------------------------|---|-----|------|
|                                | MIN   | MAX |      |
| Key press to detection delay   |   | 25  | μs   |
| Key release to detection delay |   | 25  | μs   |
| Keypad unlock timer            |   | 7   | s    |
| Keypad interrupt mask timer    |   | 31  | s    |
| Debounce                       |   | 60  | ms   |

## 6.10 Typical Characteristics

$T_A = 25^\circ\text{C}$  (unless otherwise noted)

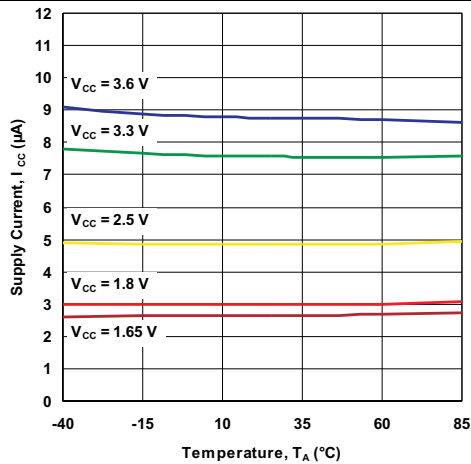


Figure 1. Supply Current vs Temperature

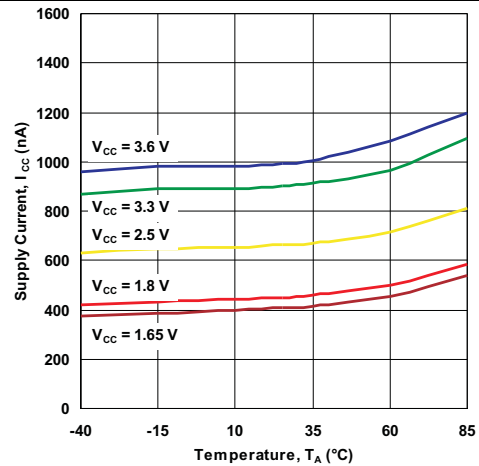


Figure 2. Standby Supply Current vs Temperature

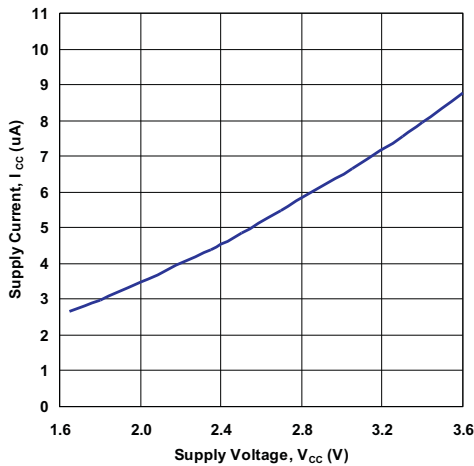


Figure 3. Supply Current vs Supply Voltage

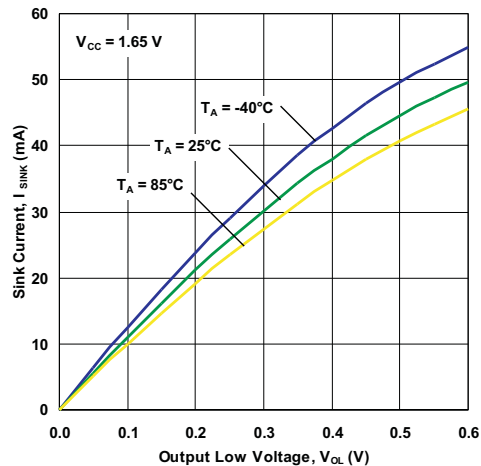


Figure 4. I/O Sink Current vs Output Low Voltage ( $V_{CC} = 1.65\text{ V}$ )

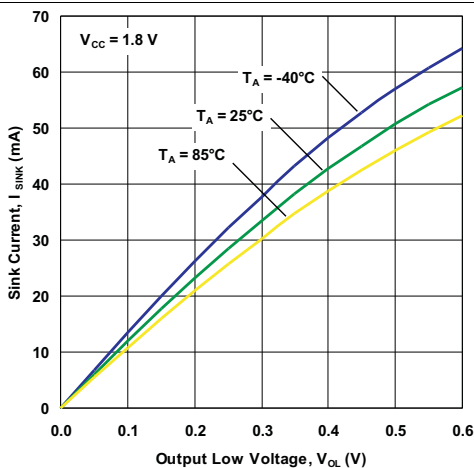


Figure 5. I/O Sink Current vs Output Low Voltage ( $V_{CC} = 1.8\text{ V}$ )

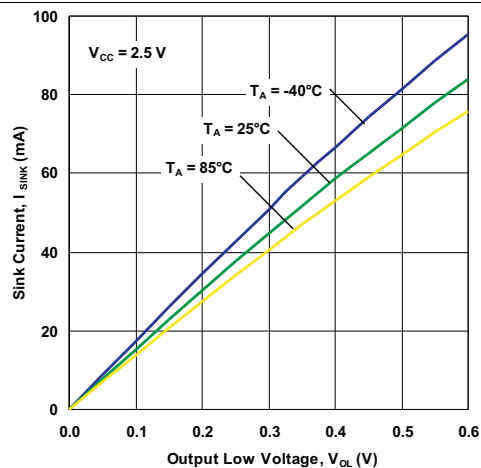


Figure 6. I/O Sink Current vs Output Low Voltage ( $V_{CC} = 2.5\text{ V}$ )

Typical Characteristics (continued)

$T_A = 25^\circ\text{C}$  (unless otherwise noted)

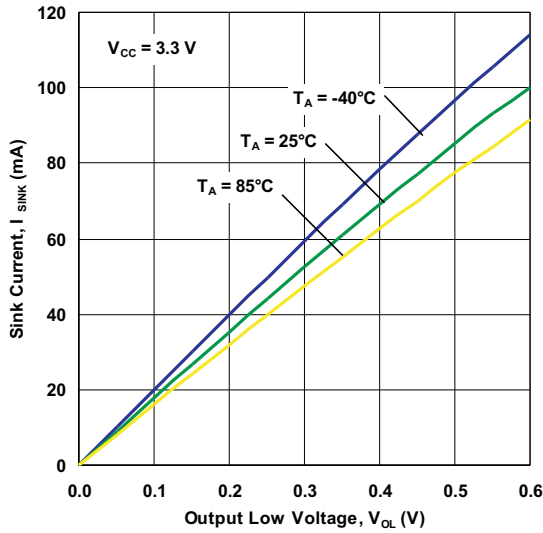


Figure 7. I/O Sink Current vs Output Low Voltage ( $V_{CC} = 3.3\text{ V}$ )

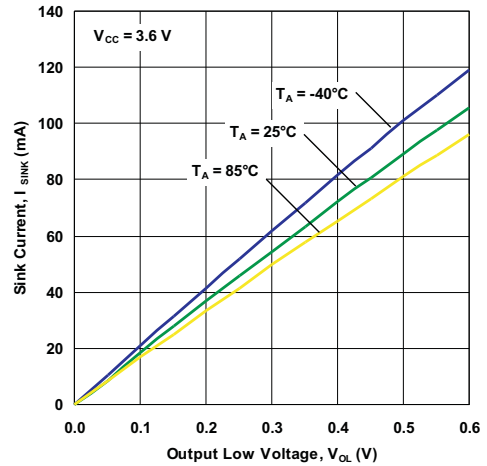


Figure 8. I/O Sink Current vs Output Low Voltage ( $V_{CC} = 3.6\text{ V}$ )

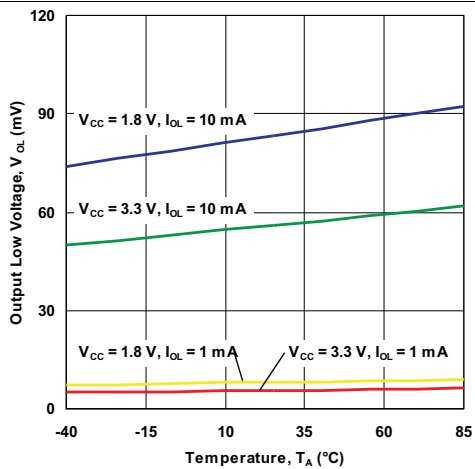


Figure 9. I/O Low Voltage vs Temperature

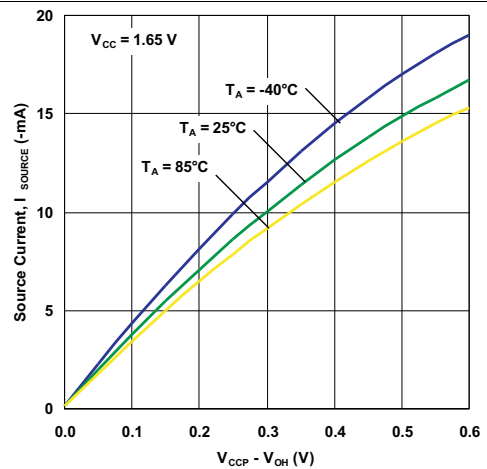


Figure 10. I/O Source Current vs Output High Voltage ( $V_{CC} = 1.65\text{ V}$ )

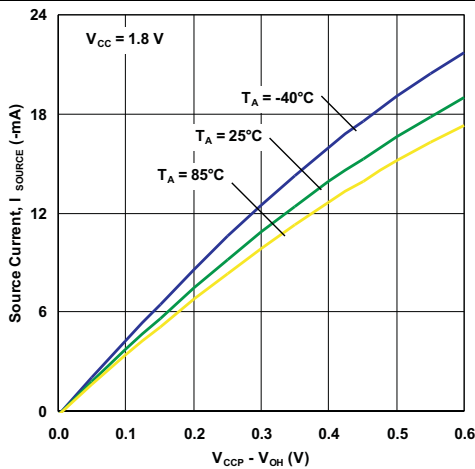


Figure 11. I/O Source Current vs Output High Voltage ( $V_{CC} = 1.8\text{ V}$ )

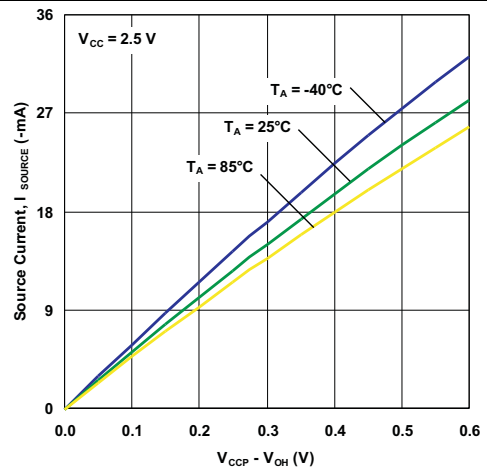
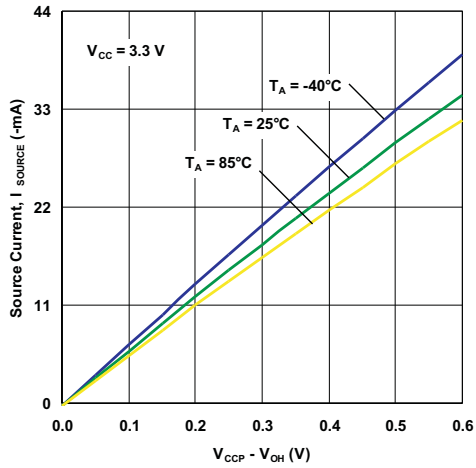


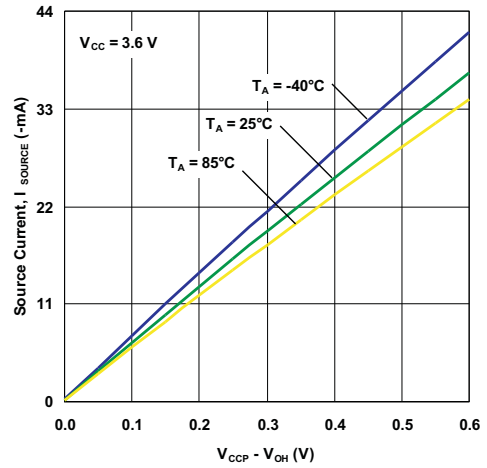
Figure 12. I/O Source Current vs Output High Voltage ( $V_{CC} = 2.5\text{ V}$ )

**Typical Characteristics (continued)**

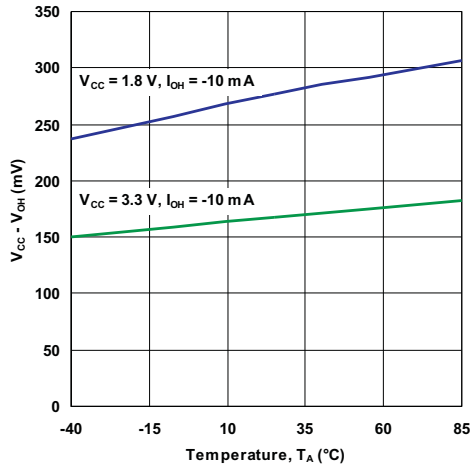
$T_A = 25^\circ\text{C}$  (unless otherwise noted)



**Figure 13. I/O Source Current vs Output High Voltage ( $V_{CC} = 3.3\text{ V}$ )**

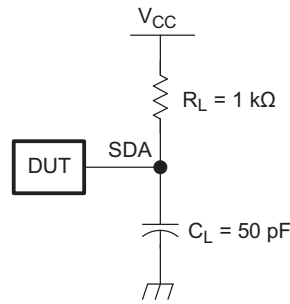


**Figure 14. I/O Source Current vs Output High Voltage ( $V_{CC} = 3.6\text{ V}$ )**

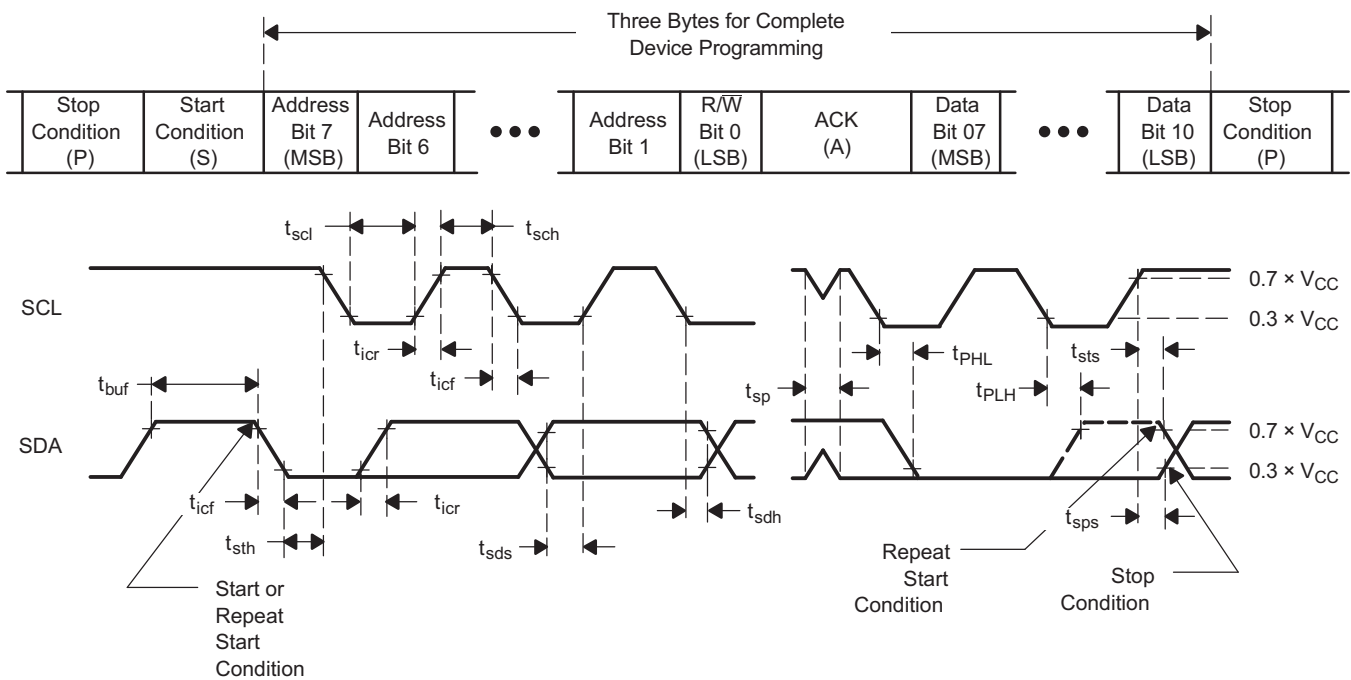


**Figure 15. I/O High Voltage vs Temperature**

## 7 Parameter Measurement Information



SDA LOAD CONFIGURATION



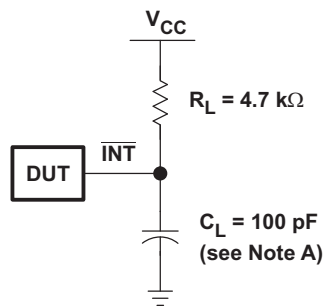
VOLTAGE WAVEFORMS

| BYTE | DESCRIPTION              |
|------|--------------------------|
| 1    | I <sup>2</sup> C address |
| 2, 3 | P-port data              |

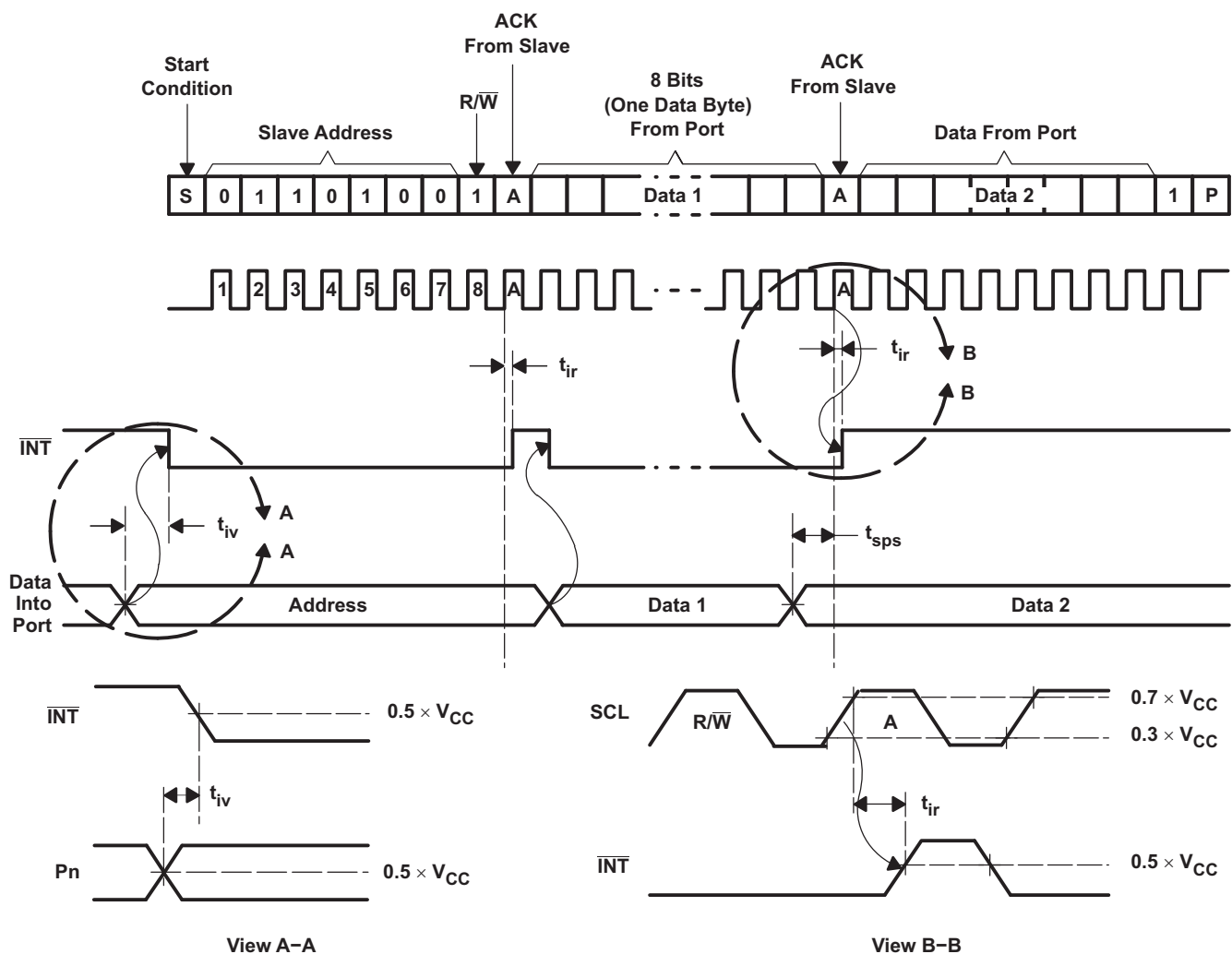
- A.  $C_L$  includes probe and jig capacitance.  $t_{ocf}$  is measured with  $C_L$  of 10 pF or 400 pF.
- B. All inputs are supplied by generators having the following characteristics: PRR  $\leq$  10 MHz,  $Z_O = 50 \Omega$ ,  $t_r/t_f \leq 30$  ns.
- C. All parameters and waveforms are not applicable to all devices.

Figure 16. I<sup>2</sup>C Interface Load Circuit And Voltage Waveforms

Parameter Measurement Information (continued)



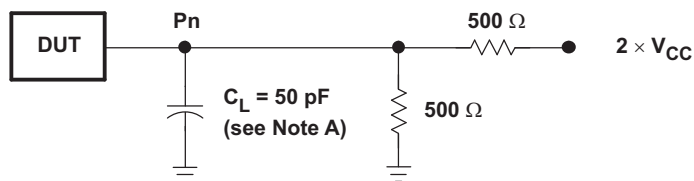
INTERRUPT LOAD CONFIGURATION



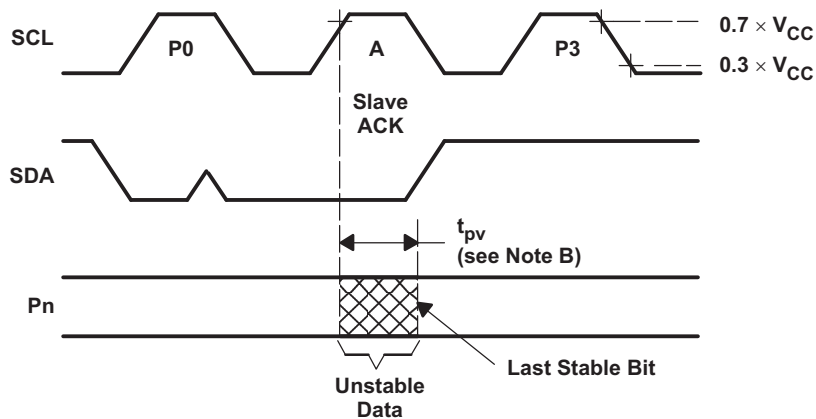
- A.  $C_L$  includes probe and jig capacitance.
- B. All inputs are supplied by generators having the following characteristics:  $PRR \leq 10 \text{ MHz}$ ,  $Z_O = 50 \Omega$ ,  $t_r/t_f \leq 30 \text{ ns}$ .
- C. All parameters and waveforms are not applicable to all devices.

Figure 17. Interrupt Load Circuit And Voltage Waveforms

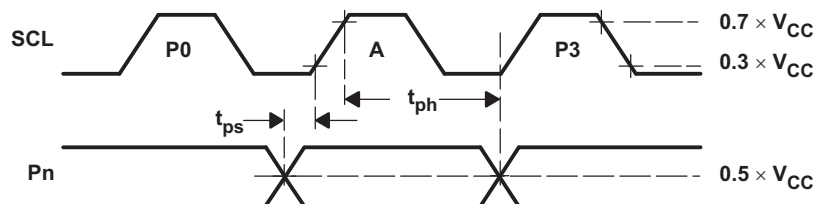
Parameter Measurement Information (continued)



P-PORT LOAD CONFIGURATION



WRITE MODE ( $R/\bar{W} = 0$ )

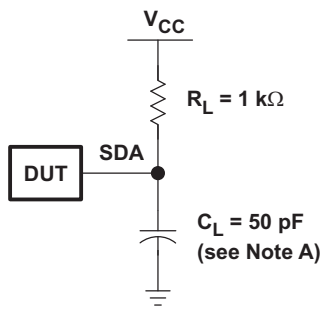


READ MODE ( $R/\bar{W} = 1$ )

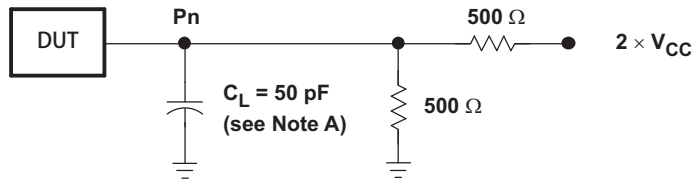
- A.  $C_L$  includes probe and jig capacitance.
- B.  $t_{pv}$  is measured from  $0.7 \times V_{CC}$  on SCL to 50% I/O ( $P_n$ ) output.
- C. All inputs are supplied by generators having the following characteristics:  $PRR \leq 10$  MHz,  $Z_O = 50 \Omega$ ,  $t_r/t_f \leq 30$  ns.
- D. The outputs are measured one at a time, with one transition per measurement.
- E. All parameters and waveforms are not applicable to all devices.

Figure 18. P Port Load Circuit And Timing Waveforms

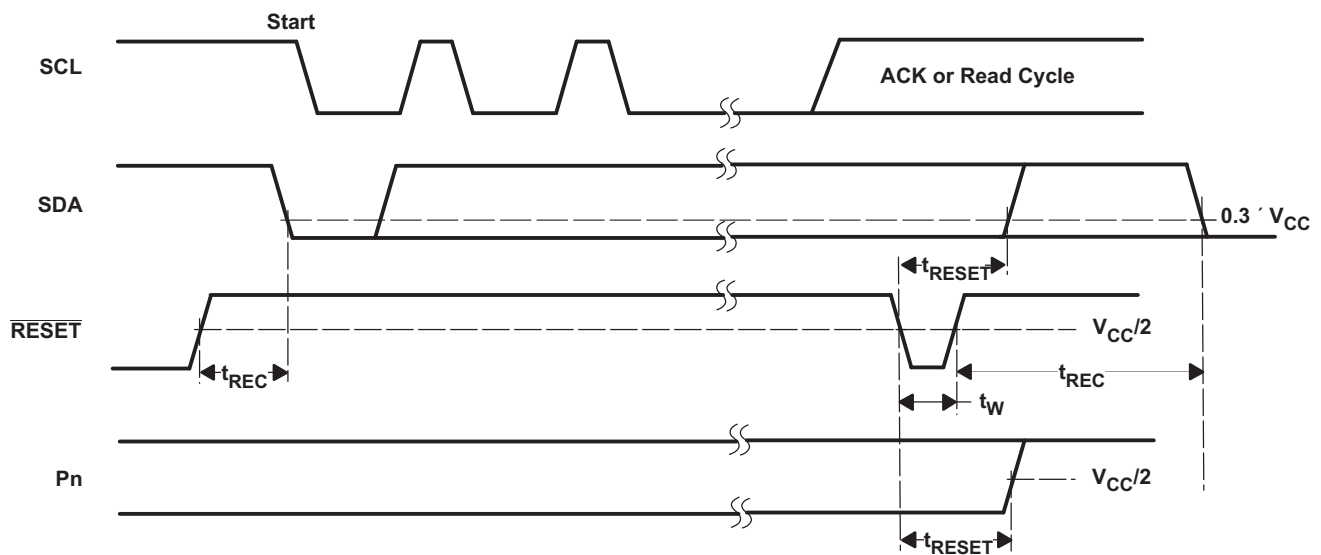
Parameter Measurement Information (continued)



SDA LOAD CONFIGURATION



P-PORT LOAD CONFIGURATION



- A.  $C_L$  includes probe and jig capacitance.
- B. All inputs are supplied by generators having the following characteristics:  $PRR \leq 10\text{ MHz}$ ,  $Z_O = 50\ \Omega$ ,  $t_r/t_f \leq 30\text{ ns}$ .
- C. The outputs are measured one at a time, with one transition per measurement.
- D. I/Os are configured as inputs.
- E. All parameters and waveforms are not applicable to all devices.

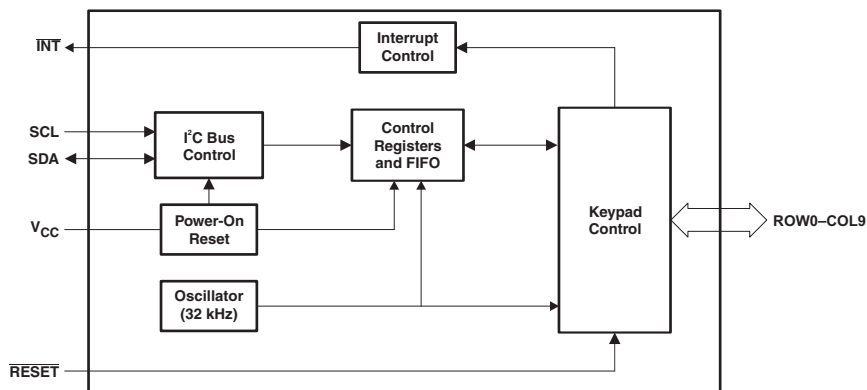
Figure 19. Reset Load Circuits And Voltage Waveforms

## 8 Detailed Description

### 8.1 Overview

The TCA8418 supports up to 10 columns Table 9 by 8 rows of keys, up to 80 keys. Any combination of these rows and columns can be configured to be added to the keypad matrix. This is done by setting the appropriate rows and columns to a value of 1 in the corresponding KP\_GPIO registers (seen in ). Once the rows and columns that are connected to the keypad matrix are added to the keypad array, then the TCA8418 will begin monitoring the keypad array, and any configured general purpose inputs (GPIs).

### 8.2 Functional Block Diagram



### 8.3 Feature Description

#### 8.3.1 Key Events

##### 8.3.1.1 Key Event Table

The TCA8418 can be configured to support many different configurations of keypad setups. All 18 GPIOs for the rows and columns can be used to support up to 80 keys in a key pad array. Another option is that all 18 GPIOs be used for GPIs to read 18 buttons which are not connected in an array. Any combination in between is also acceptable (for example, a 3 x 4 keypad matrix and using the remaining 11 GPIOs as a combination of inputs and outputs).

For both types of inputs (keypad matrix and a GPI), a key event can be added to the key event FIFO. The values that are added to the FIFO depend on the configuration (keypad array or GPI) and on which port the press was read on. The tables below show the values that correspond to both types of configurations.

Key values below are represented in decimal values, because the 10s place is used to mark the row, and the ones place is used to denote the column. It is more clear to see the numbering convention used when viewed in decimal values.

**Table 1. Key Event Table (Keypad Array)**

|    | 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 |

**Table 2. Key Event Table (Row GPI Events)**

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

**Table 3. Key Event Table (Column GPI Events)**

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

### 8.3.1.2 General Purpose Input (GPI) Events

A column or row configured as GPI can be programmed to be part of the [Key Event Table](#), hence becomes also capable of generating Key Event Interrupt. A Key Event Interrupt caused by a GPI follow the same process flow as a Key Event Interrupt caused by a Key press.

GPIs configured as part of the Key Event Table allows for single key switches to be monitored as well as other GPI interrupts. As part of the Event Table, GPIs are represented with decimal value of 97 and run through decimal value of 114. R0-R7 are represented by 97-104 and C0-C9 are represented by 105-114

For a GPI that is set as active high, and is enabled in the Key Event Table, the state-machine will add an event to the event count and event table whenever that GPI goes high. If the GPI is set to active low, a transition from high to low will be considered a press and will also be added to the event count and event table. Once the interrupt state has been met, the state machine will internally set an interrupt for the opposite state programmed in the register to avoid polling for the released state, hence saving current. Once the released state is achieved, it will add it to the event table. The press and release will still be indicated by bit 7 in the event register.

The GPI Events can also be used as unlocked sequences. When the GPI\_EM bit is set, GPI events will not be tracked when the keypad is locked. GPI\_EM bit must be cleared for the GPI events to be tracked in the event counter and table when the keypad is locked.

### 8.3.1.3 Key Event (FIFO) Reading

The TCA8418 has a 10-byte event FIFO, which stores any key presses or releases which have been configured to be added to the Key Event Table. All ROWs and COLs added to the keypad matrix via the [KP\\_GPIO1-3 Registers](#) will have any key pad events added to the FIFO. Any GPIs configured with a 1 in the [GPI\\_EM1-3 Registers](#) will also be part of the event FIFO.

When the host wishes to read the FIFO, the following procedure is recommended.

1. Read the [INT\\_STAT \(0x02\)](#) register to determine what asserted the  $\overline{\text{INT}}$  line. If GPI\_INT or K\_INT is set, then a key event has occurred, and the event is stored in the FIFO.
2. Read the [KEY\\_LCK\\_EC \(0x03\)](#) register, bits [3:0] to see how many events are stored in FIFO.
3. Read the [KEY\\_EVENT\\_A \(0x04\)](#) register. Bit 7 value '0' signifies key release, value 1 signifies key press. Bits [6:0] state which key was pressed with respect to the [Key Event Table](#). With each read of the key event register, the event counter in KEY\_LCK\_EC[3:0] will decrease by 1, and the FIFO will shift the events down 1 register.
4. Repeat step 3 until either KEY\_LCK\_EC[3:0] = 0 or KEY\_EVENT\_A = 0. This signifies that the FIFO is empty.
5. Reset the [INT\\_STAT](#) interrupt flag which was causing the interrupt by writing a 1 to the specific bit.

As an example, consider the following key presses.

**Table 4. Example Key Sequence**

| Event # | Key (Decimal Value) | Press/Release |
|---------|---------------------|---------------|
| 1       | 1                   | Press         |
| 2       | 32                  | Press         |
| 3       | 1                   | Release       |
| 4       | 32                  | Release       |
| 5       | 23                  | Press         |
| 6       | 23                  | Release       |
| 7       | 45                  | Press         |
| 8       | 41                  | Press         |
| 9       | 41                  | Release       |
| 10      | 45                  | Release       |

If this example key sequence occurs, then while performing the recommended read procedure listed above, the host would see the following information. Information at the top of the list is of an initial read to the KEY\_LCK\_EC[3:0] register.

**Table 5. Example Key Sequence**

| KEY_LCK_EC[3:0] Value | KEY_EVENT_A Value (Binary/Hex) | Key (Decimal Value) | Press/Release |
|-----------------------|--------------------------------|---------------------|---------------|
| 10                    | N/A                            | N/A                 | N/A           |
| 9                     | 1 000 0001 (0x81)              | 1                   | Press         |
| 8                     | 1 010 0000 (0xA0)              | 32                  | Press         |
| 7                     | 0 000 0001 (0x01)              | 1                   | Release       |
| 6                     | 0 010 0000 (0x20)              | 32                  | Release       |
| 5                     | 1 001 0111 (0x97)              | 23                  | Press         |
| 4                     | 0 001 0111 (0x17)              | 23                  | Release       |
| 3                     | 1 010 1101 (0xAD)              | 45                  | Press         |
| 2                     | 1 010 1001 (0xA9)              | 41                  | Press         |
| 1                     | 0 010 1001 (0x29)              | 41                  | Release       |
| 0                     | 0 010 1101 (0x2D)              | 45                  | Release       |

#### <sup>(1)</sup>8.3.1.4 Key Event Overflow

The TCA8418 has the ability to handle an overflow of the key event FIFO. An overflow event occurs when the FIFO is full of events (10 key events are stored) and a new key event occurs. In short, this means that the TCA8418 does not have the ability to hold any more key press information in the internal buffer. When this occurs, the OVR\_FLOW\_INT bit in the *INT\_STAT Register* is set, and if the OVR\_FLOW\_IEN bit is set in the *CFG Register*, then the  $\overline{\text{INT}}$  output will be asserted low to let the processor know that an overflow has occurred.

The TCA8418 has the ability to handle an overflow in 1 of two ways, which is determined by the bit value of the OVR\_FLOW\_M bit in the *CFG Register*.

Please see the *Overflow Errata* section for more information about overflow behavior.

**Table 6. OVR\_FLOW\_M Bit**

| OVR_FLOW_M Value | Overflow Mode      | Behavior   |
|------------------|--------------------|--|
| 1                | Enabled            | Overflow data shifts with last event pushing first event out |
| 0                | Disabled (Default) | Overflow data is not stored and lost                         |

(1) Note that the MSB of the KEY\_EVENT\_A value signifies a key press or release. MSB Bit = 1 signifies a key press; MSB Bit = 0 signifies a key release

Consider the example below, if the FIFO is full of the key presses and a new key press comes in. This new overflow key press will be a key press of key 2 (0x82 is the hex representation of a key 2 press event)

**Table 7. Key Event Overflow Handling**

| FIFO Register | Original Value | After Key 1 Press Event (0x82) |                |
|---------------|----------------|--------------------------------|----------------|
|               |                | OVR_FLOW_M = 1                 | OVR_FLOW_M = 0 |
| A             | 0x81           | 0xA0                           | 0x81           |
| B             | 0xA0           | 0x01                           | 0xA0           |
| C             | 0x01           | 0x20                           | 0x01           |
| D             | 0x20           | 0x97                           | 0x20           |
| E             | 0x97           | 0x17                           | 0x97           |
| F             | 0x17           | 0xAD                           | 0x17           |
| H             | 0xAD           | 0xA9                           | 0xAD           |
| I             | 0xA9           | 0x29                           | 0xA9           |
| J             | 0x29           | 0x2D                           | 0x29           |
| K             | 0x2D           | <b>0x82</b>                    | 0x2D           |

### 8.3.2 Keypad Lock/Unlock

This user can lock the keypad through the lock/unlock feature in this device. Once the keypad is locked by setting BIT6 in [KEY\\_LCK\\_EC](#), it can prevent the generation of key event interrupts and recorded key events. The unlock keys can be programmed with any value of the keys in the keypad matrix or any general purpose input (GPI) values that are part of the [Key Event Table](#). When the keypad lock interrupt mask timer is non-zero, the user will need to press two specific keys before an keylock interrupt is generated or keypad events are recorded. A key event interrupt is generated the first time a user presses any key. This first interrupt can be used to turn on an LCD and display the unlock message. The processor will then read the lock status register to see if the keypad is unlocked. The next interrupt (keylock interrupt) will not be generated unless both unlock keys sequences are correct. If correct Unlock keys are not pressed before the mask timer expires, the state machine will start over again.

The recommended procedure to lock the keypad is to do the following

1. Determine which keys will be used for the unlock sequence. The key value from the [Key Event Tables](#) needs to be entered into the [UNLOCK1](#) and [UNLOCK2](#) registers.
2. The UNLOCK1 to UNLOCK2 timer duration must be set by entering the desired seconds (valid range is 0 to 7 seconds) into bits [2:0] of the [KP\\_LCK\\_TMR](#) register.
3. If an interrupt mask is desired (see [Keypad Lock Interrupt Mask Timer](#)), then the desired interrupt mask duration (valid range is 0 to 31 seconds) must be entered into bits [7:3] of the [KP\\_LCK\\_TMR](#) register.
4. When the host is ready to lock the keypad, a 1 is to be written to the K\_LCK\_EN bit (BIT6) in the [KEY\\_LCK\\_EC](#) register. This will lock the keypad.
5. If the host wishes to manually unlock the keypad, writing a '0' to the K\_LCK\_EN bit (BIT6) in the [KEY\\_LCK\\_EC](#) register will unlock the keypad.

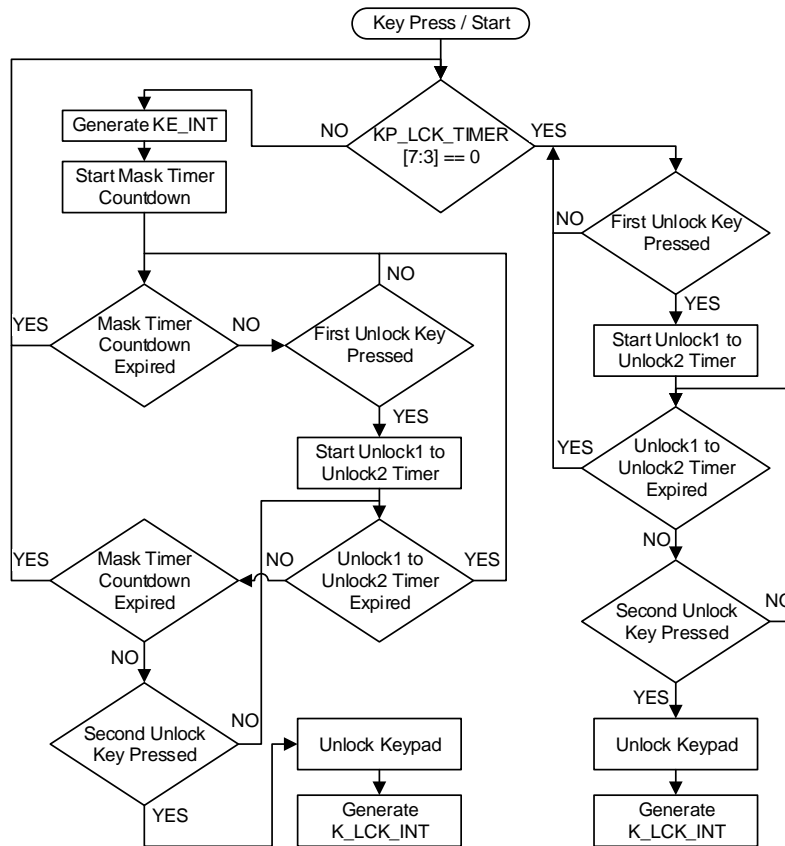


Figure 20. Keypad Lock Flowchart

### 8.3.3 Keypad Lock Interrupt Mask Timer

The TCA8418 features a *Keypad Lock/Unlock* feature which allows the user to stop the generation of key event interrupts by locking the keypad. There is an interrupt mask timer feature with the keypad lock, which allows the generation of a single interrupt when a key is pressed, primarily for the purpose of LCD backlighting. Note that this interrupt mask timer can also be used to limit the number of interrupts generated for a given amount of time.

The interrupt mask timer is enabled by setting bits [7:3] of the *KP\_LCK\_TIMER* register. The value in this register can be anywhere from 0 to 31 seconds (note that a value of 0 will disable this interrupt mask feature). When a keypad is locked and the interrupt mask timer is set to a non-zero value, this will enable the interrupt mask timer. This interrupt mask timer limits the amount of interrupts generated. Typically, this is used with the *Keypad Lock/Unlock* feature for LCD back lights. It is easiest to explain this feature with the following example; A mobile device has a LCD screen with a back light display which turns off after 10 seconds to save power. Normally, an interrupt to the processor would tell this LCD back light to turn on. When the keypad is locked, no interrupts are generated, so the back light will never turn on. This is where the interrupt mask feature is used. Please refer to [Figure 20](#). The procedure for an example is below.

1. Since the back light turns off after 10 seconds of no interrupts, the interrupt mask timer (*KP\_LCK\_TIMER*[7:3]) gets set to 10 seconds. Keypad is then locked.
2. When the first key press is detected, the TCA8418 sends an interrupt to the processor and starts a 10 second count down.
3. If the correct unlock sequence is not entered within the 10 seconds, no interrupts are sent and the back light will turn off.
4. After the 10 second timer has expired, if another key press occurs while keypad is locked (regardless of whether it is a correct unlock key or not), another interrupt is generated and the 10 second count down begins again.

### 8.3.4 Control-Alt-Delete Support

The TCA8418 can support normal key presses, but it also can support a <Ctrl><Alt><Del> (CAD) key press. This feature allows the host to recognize a specific key press and alert the host that the combination has occurred. The TCA8418 will recognize a <Ctrl><Alt><Del> key press if keys 1, 11, and 21 are all pressed at the same time. These keys are referenced to the key values listed in the [Key Event Table](#). Note that this key combination that triggers a CAD interrupt is not adjustable, and must be keys 1, 11, and 21.

Please see [CAD Interrupt Errata](#) for more information.

### 8.3.5 Interrupt Output

An interrupt is generated by any rising or falling edge of the port inputs in the input mode. After time  $t_{iv}$ , the signal  $\overline{INT}$  is valid. Resetting the interrupt circuit is achieved when data on the port is changed to the original setting or data is read from the port that generated the interrupt. Resetting occurs in the read mode at the acknowledge (ACK) or not acknowledge (NACK) bit after the rising edge of the SCL signal. Interrupts that occur during the ACK or NACK clock pulse can be lost (or be very short) due to the resetting of the interrupt during this pulse. Each change of the I/Os after resetting is detected and is transmitted as  $\overline{INT}$ .

Reading from or writing to another device does not affect the interrupt circuit, and a pin configured as an output cannot cause an interrupt. Changing an I/O from an output to an input may cause a false interrupt to occur, if the state of the pin does not match the contents of the input port register.

The  $\overline{INT}$  output has an open-drain structure and requires a pull-up resistor to  $V_{CC}$  depending on the application. If the  $\overline{INT}$  signal is connected back to the processor that provides the SCL signal to the TCA8418, then the  $\overline{INT}$  pin has to be connected to  $V_{CC}$ . If not, the  $\overline{INT}$  pin can be connected to  $V_{CC}$ .

#### 8.3.5.1 50 Micro-second Interrupt Configuration

The TCA8418 provides the capability of deasserting the interrupt for 50  $\mu$ 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 50  $\mu$ s deassertion. When the INT\_CFG bit is cleared,  $\overline{INT}$  remains asserted if the host tries to clear the interrupt. This feature is particularly useful for software development and edge triggering applications.

## 8.4 Device Functional Modes

### 8.4.1 Power-On Reset (POR)

When power (from 0 V) is applied to  $V_{CC}$ , an internal power-on reset circuit holds the TCA8418 in a reset condition until  $V_{CC}$  has reached  $V_{PORR}$ . At that time, the reset condition is released, and the TCA8418 registers and I<sup>2</sup>C/SMBus state machine initialize to their default states. After that,  $V_{CC}$  must be lowered to below  $V_{PORF}$  and back up to the operating voltage for a power-reset cycle. See [Power Supply Recommendations](#) for more information on power up reset requirements.

### 8.4.2 Powered (Key Scan Mode)

The TCA8418 can be used to read GPI from single buttons, or configured in key scan mode to read an array of keys. In key scan mode, there are two modes of operation.

#### 8.4.2.1 Idle Key Scan Mode

Once the TCA8418 has had the keypad array configured, it will enter idle mode when no keys are being pressed. All columns configured as part of the keypad array will be driven low and all rows configured as part of the keypad array will be set to inputs, with pull-up resistors enabled. During idle mode, the internal oscillator is turned off so that power consumption is low as the device awaits a key press.

#### 8.4.2.2 Active Key Scan Mode

When the TCA8418 is in idle key scan mode, the device awaits a key press. Once a key is pressed in the array, a low signal on one of the ROW pin inputs triggers an interrupt, which will turn on the internal oscillator and enter the active key scan mode. At this point, the TCA8418 will start the key scan algorithm to determine which key is being pressed, and/or it will use the internal oscillator for debouncing. Once all keys have been released, the device will enter idle key scan mode.

## 8.5 Programming

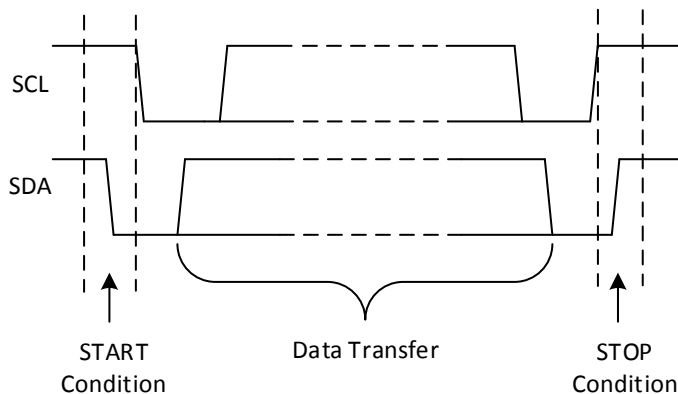
### 8.5.1 I<sup>2</sup>C Interface

The TCA8418 has a standard bidirectional I<sup>2</sup>C interface that is controlled by a master device in order to be configured or read the status of this device. Each slave on the I<sup>2</sup>C bus has a specific device address to differentiate between other slave devices that are on the same I<sup>2</sup>C bus. Many slave devices will require configuration upon startup to set the behavior of the device. This is typically done when the master accesses internal register maps of the slave, which have unique register addresses. A device can have one or multiple registers where data is stored, written, or read.

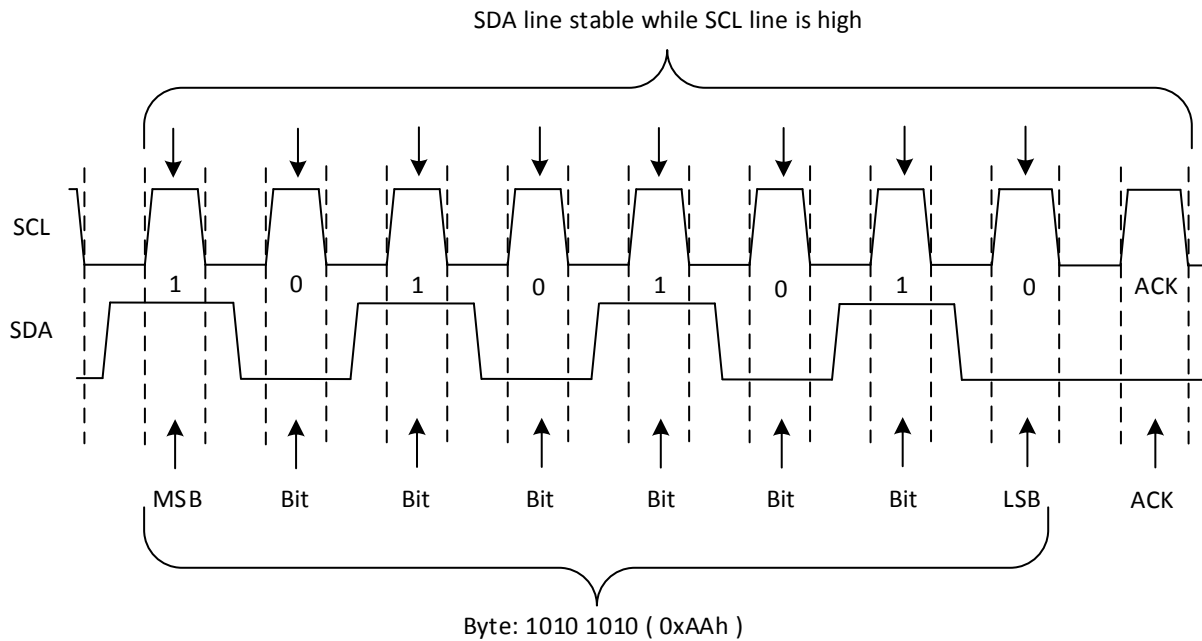
The physical I<sup>2</sup>C interface consists of the serial clock (SCL) and serial data (SDA) lines. Both SDA and SCL lines must be connected to V<sub>CC</sub> through a pull-up resistor. The size of the pull-up resistor is determined by the amount of capacitance on the I<sup>2</sup>C lines. (For further details, refer to *I<sup>2</sup>C pull-up Resistor Calculation (SLVA689)*.) Data transfer may be initiated only when the bus is idle. A bus is considered idle if both SDA and SCL lines are high after a STOP condition.

The following is the general procedure for a master to access a slave device:

1. If a master wants to send data to a slave:
  - Master-transmitter sends a START condition and addresses the slave-receiver.
  - Master-transmitter sends data to slave-receiver.
  - Master-transmitter terminates the transfer with a STOP condition.
2. If a master wants to receive or read data from a slave:
  - Master-receiver sends a START condition and addresses the slave-transmitter.
  - Master-receiver sends the requested register to read to slave-transmitter.
  - Master-receiver receives data from the slave-transmitter.
  - Master-receiver terminates the transfer with a STOP condition.



**Figure 21. Definition of Start and Stop Conditions**

**Programming (continued)**

**Figure 22. Bit Transfer**
**8.5.2 Bus Transactions**

Data must be sent to and received from the slave devices, and this is accomplished by reading from or writing to registers in the slave device.

Registers are locations in the memory of the slave which contain information, whether it be the configuration information or some sampled data to send back to the master. The master must write information to these registers in order to instruct the slave device to perform a task.

While it is common to have registers in I<sup>2</sup>C slaves, note that not all slave devices will have registers. Some devices are simple and contain only 1 register, which may be written to directly by sending the register data immediately after the slave address, instead of addressing a register. An example of a single-register device would be an 8-bit I<sup>2</sup>C switch, which is controlled via I<sup>2</sup>C commands. Since it has 1 bit to enable or disable a channel, there is only 1 register needed, and the master merely writes the register data after the slave address, skipping the register number.

**8.5.2.1 Writes**

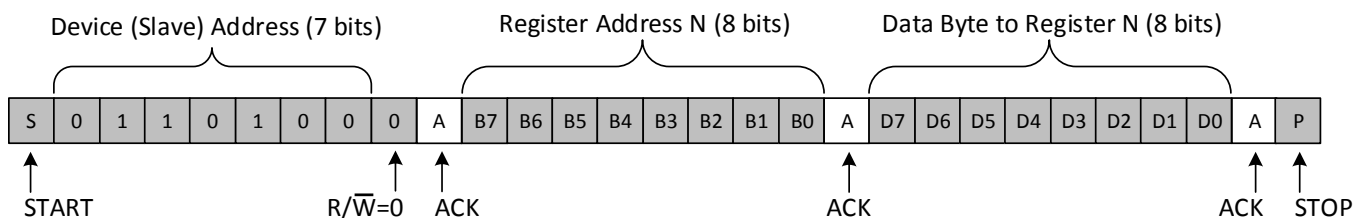
To write on the I<sup>2</sup>C bus, the master will send a START condition on the bus with the address of the slave, as well as the last bit (the R/W bit) set to 0, which signifies a write. After the slave sends the acknowledge bit, the master will then send the register address of the register to which it wishes to write. The slave will acknowledge again, letting the master know it is ready. After this, the master will start sending the register data to the slave until the master has sent all the data necessary (which is sometimes only a single byte), and the master will terminate the transmission with a STOP condition.

Figure 23 shows an example of writing a single byte to a register.

**Programming (continued)**

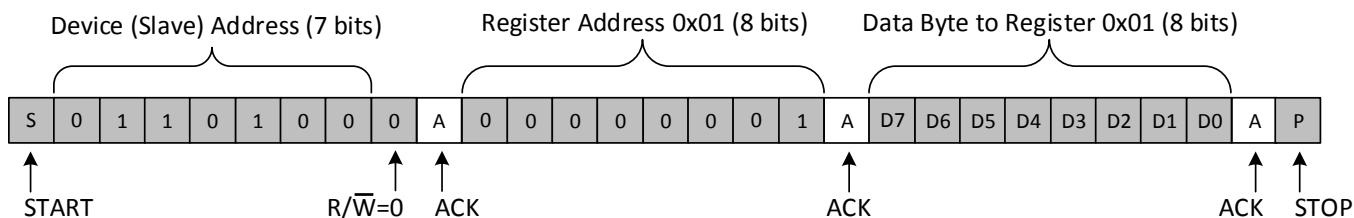
- Master controls SDA line
- Slave controls SDA line

Write to one register in a device



**Figure 23. Write to Register**

- Master controls SDA line
- Slave controls SDA line



**Figure 24. Write to Configuration Register**

## Programming (continued)

### 8.5.2.2 Reads

Reading from a slave is very similar to writing, but requires some additional steps. In order to read from a slave, the master must first instruct the slave which register it wishes to read from. This is done by the master starting off the transmission in a similar fashion as the write, by sending the address with the R/W bit equal to 0 (signifying a write), followed by the register address it wishes to read from. Once the slave acknowledges this register address, the master will send a START condition again, followed by the slave address with the R/W bit set to 1 (signifying a read). This time, the slave will acknowledge the read request, and the master will release the SDA bus but will continue supplying the clock to the slave. During this part of the transaction, the master will become the master-receiver, and the slave will become the slave-transmitter.

The master will continue to send out the clock pulses, but will release the SDA line so that the slave can transmit data. At the end of every byte of data, the master will send an ACK to the slave, letting the slave know that it is ready for more data. Once the master has received the number of bytes it is expecting, it will send a NACK, signaling to the slave to halt communications and release the bus. The master will follow this up with a STOP condition.

Figure 25 shows an example of reading a single byte from a slave register.

- Master controls SDA line
- Slave controls SDA line

#### Read from one register in a device

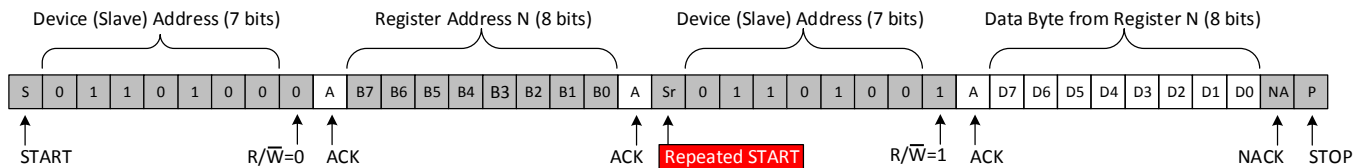


Figure 25. Read from Register

## 8.6 Register Maps

### 8.6.1 Device Address

The address of the TCA8418 is shown in [Table 8](#).

**Table 8. TCA8418 Device Addresses**

| BYTE                           | BIT     |   |   |   |   |   |   |         |
|--------------------------------|---------|---|---|---|---|---|---|---------|
|                                | 7 (MSB) | 6 | 5 | 4 | 3 | 2 | 1 | 0 (LSB) |
| I <sup>2</sup> C slave address | 0       | 1 | 1 | 0 | 1 | 0 | 0 | R/W     |

The last bit of the slave address defines the operation (read or write) to be performed. A high (1) selects a read operation, while a low (0) selects a write operation.

### 8.6.2 Control Register and Command Byte

Following the successful acknowledgment of the address byte, the bus master sends a command byte, which is stored in the control register in the TCA8418. The command byte indicates the register that will be updated with information. All registers can be read and written to by the system master.

[Table 9](#) shows all the registers within this device and their descriptions. The default value in all registers is 0.

**Table 9. Register Descriptions**

| ADDRESS | REGISTER NAME | REGISTER DESCRIPTION   | 7         | 6         | 5           | 4         | 3              | 2          | 1         | 0         |
|---------|---------------|--|-----------|-----------|-------------|-----------|----------------|------------|-----------|-----------|
| 0x00    | Reserved      | Reserved   |           |           |             |           |                |            |           |           |
| 0x01    | CFG           | Configuration register (interrupt processor interrupt enables) | AI        | GPI_E_CGF | OVR_FL_OW_M | INT_CFG   | OVR_F LOW_I EN | K_LC K_IEN | GPI_IEN   | KE_IEN    |
| 0x02    | INT_STAT      | Interrupt status register                                      | N/A<br>0  | N/A<br>0  | N/A<br>0    | CAD_I NT  | OVR_F LOW_I NT | K_LC K_INT | GPI_INT   | K_INT     |
| 0x03    | KEY_LCK_EC    | Key lock and event counter register                            | N/A<br>0  | K_LCK_EN  | LCK2        | LCK1      | KLEC3          | KLEC2      | KLEC1     | KLEC0     |
| 0x04    | KEY_EVENT_A   | Key event register A   | KEA7<br>0 | KEA6<br>0 | KEA5<br>0   | KEA4<br>0 | KEA3<br>0      | KEA2<br>0  | KEA1<br>0 | KEA0<br>0 |
| 0x05    | KEY_EVENT_B   | Key event register B   | KEB7<br>0 | KEB6<br>0 | KEB5<br>0   | KEB4<br>0 | KEB3<br>0      | KEB2<br>0  | KEB1<br>0 | KEB0<br>0 |
| 0x06    | KEY_EVENT_C   | Key event register C   | KEC7<br>0 | KEC6<br>0 | KEC5<br>0   | KEC4<br>0 | KEC3<br>0      | KEC2<br>0  | KEC1<br>0 | KEC0<br>0 |
| 0x07    | KEY_EVENT_D   | Key event register D   | KED7<br>0 | KED6<br>0 | KED5<br>0   | KED4<br>0 | KED3<br>0      | KED2<br>0  | KED1<br>0 | KED0<br>0 |
| 0x08    | KEY_EVENT_E   | Key event register E   | KEE7<br>0 | KEE6<br>0 | KEE5<br>0   | KEE4<br>0 | KEE3<br>0      | KEE2<br>0  | KEE1<br>0 | KEE0<br>0 |
| 0x09    | KEY_EVENT_F   | Key event register F   | KEF7<br>0 | KEF6<br>0 | KEF5<br>0   | KEF4<br>0 | KEF3<br>0      | KEF2<br>0  | KEF1<br>0 | KEF0<br>0 |
| 0x0A    | KEY_EVENT_G   | Key event register G   | KEG7<br>0 | KEG6<br>0 | KEG5<br>0   | KEG4<br>0 | KEG3<br>0      | KEG2<br>0  | KEG1<br>0 | KEG0<br>0 |
| 0x0B    | KEY_EVENT_H   | Key event register H   | KEH7<br>0 | KEH6<br>0 | KEH5<br>0   | KEH4<br>0 | KEH3<br>0      | KEH2<br>0  | KEH1<br>0 | KEH0<br>0 |
| 0x0C    | KEY_EVENT_I   | Key event register I   | KEI7<br>0 | KEI6<br>0 | KEI5<br>0   | KEI4<br>0 | KEI3<br>0      | KEI2<br>0  | KEI1<br>0 | KEI0<br>0 |
| 0x0D    | KEY_EVENT_J   | Key event register J   | KEJ7<br>0 | KEJ6<br>0 | KEJ5<br>0   | KEJ4<br>0 | KEJ3<br>0      | KEJ2<br>0  | KEJ1<br>0 | KEJ0<br>0 |
| 0x0E    | KP_LCK_TIMER  | Keypad lock 1 to lock 2 timer                                  | KL7       | KL6       | KL5         | KL4       | KL3            | KL2        | KL1       | KL0       |
| 0x0F    | UNLOCK1       | Unlock key 1   | UK1_7     | UK1_6     | UK1_5       | UK1_4     | UK1_3          | UK1_2      | UK1_1     | UK1_0     |
| 0x10    | UNLOCK1       | Unlock key2  | UK2_7     | UK2_6     | UK2_5       | UK2_4     | UK2_3          | UK2_2      | UK2_1     | UK2_0     |

**Table 9. Register Descriptions (continued)**

| ADDRESS | REGISTER NAME                           | REGISTER DESCRIPTION                                | 7         | 6         | 5         | 4         | 3         | 2         | 1         | 0         |
|---------|---|---|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|
| 0x11    | GPIO_INT_STAT1                          | GPIO interrupt status                               | R7IS<br>0 | R6IS<br>0 | R5IS<br>0 | R4IS<br>0 | R3IS<br>0 | R2IS<br>0 | R1IS<br>0 | R0IS<br>0 |
| 0x12    | GPIO_INT_STAT2                          | GPIO interrupt status                               | C7IS<br>0 | C6IS<br>0 | C5IS<br>0 | C4IS<br>0 | C3IS<br>0 | C2IS<br>0 | C1IS<br>0 | C0IS<br>0 |
| 0x13    | GPIO_INT_STAT3                          | GPIO interrupt status                               | N/A<br>0  | N/A<br>0  | N/A<br>0  | N/A<br>0  | N/A<br>0  | N/A<br>0  | C9IS<br>0 | C8IS<br>0 |
| 0x14    | GPIO_DAT_STAT1<br>(read twice to clear) | GPIO data status                                    | R7DS      | R6DS      | R5DS      | R4DS      | R3DS      | R2DS      | R1DS      | R0DS      |
| 0x15    | GPIO_DAT_STAT2<br>(read twice to clear) | GPIO data status                                    | C7DS      | C6DS      | C5DS      | C4DS      | C3DS      | C2DS      | C1DS      | C0DS      |
| 0x16    | GPIO_DAT_STAT3<br>(read twice to clear) | GPIO data status                                    | N/A<br>0  | N/A<br>0  | N/A<br>0  | N/A<br>0  | N/A<br>0  | N/A<br>0  | C9DS      | C8DS      |
| 0x17    | GPIO_DAT_OUT1                           | GPIO data out                                       | R7DO<br>0 | R6DO<br>0 | R5DO<br>0 | R4DO<br>0 | R3DO<br>0 | R2DO<br>0 | R1DO<br>0 | R0DO<br>0 |
| 0x18    | GPIO_DAT_OUT2                           | GPIO data out                                       | C7DO<br>0 | C6DO<br>0 | C5DO<br>0 | C4DO<br>0 | C3DO<br>0 | C2DO<br>0 | C1DO<br>0 | C0DO<br>0 |
| 0x19    | GPIO_DAT_OUT3                           | GPIO data out                                       | N/A<br>0  | N/A<br>0  | N/A<br>0  | N/A<br>0  | N/A<br>0  | N/A<br>0  | C9DO<br>0 | C8DO<br>0 |
| 0x1A    | GPIO_INT_EN1                            | GPIO interrupt enable                               | R7IE<br>0 | R6IE<br>0 | R5IE<br>0 | R4IE<br>0 | R3IE<br>0 | R2IE<br>0 | R1IE<br>0 | R0IE<br>0 |
| 0x1B    | GPIO_INT_EN2                            | GPIO interrupt enable                               | C7IE<br>0 | C6IE<br>0 | C5IE<br>0 | C4IE<br>0 | C3IE<br>0 | C2IE<br>0 | C1IE<br>0 | C0IE<br>0 |
| 0x1C    | GPIO_INT_EN3                            | GPIO interrupt enable                               | N/A<br>0  | N/A<br>0  | N/A<br>0  | N/A<br>0  | N/A<br>0  | N/A<br>0  | C9IE<br>0 | C8IE<br>0 |
| 0x1D    | KP_GPIO1                                | Keypad or GPIO selection<br>0: GPIO<br>1: KP matrix | ROW7<br>0 | ROW6<br>0 | ROW5<br>0 | ROW4<br>0 | ROW3<br>0 | ROW2<br>0 | ROW1<br>0 | ROW0<br>0 |
| 0x1E    | KP_GPIO2                                | Keypad or GPIO selection<br>0: GPIO<br>1: KP matrix | COL7<br>0 | COL6<br>0 | COL5<br>0 | COL4<br>0 | COL3<br>0 | COL2<br>0 | COL1<br>0 | COL0<br>0 |
| 0x1F    | KP_GPIO3                                | Keypad or GPIO selection<br>0: GPIO<br>1: KP matrix | N/A<br>0  | N/A<br>0  | N/A<br>0  | N/A<br>0  | N/A<br>0  | N/A<br>0  | COL9<br>0 | COL8<br>0 |
| 0x20    | GPI_EM1                                 | GPI event mode 1                                    | ROW7<br>0 | ROW6<br>0 | ROW5<br>0 | ROW4<br>0 | ROW3<br>0 | ROW2<br>0 | ROW1<br>0 | ROW0<br>0 |
| 0x21    | GPI_EM2                                 | GPI event mode 2                                    | COL7<br>0 | COL6<br>0 | COL5<br>0 | COL4<br>0 | COL3<br>0 | COL2<br>0 | COL1<br>0 | COL0<br>0 |
| 0x22    | GPI_EM3                                 | GPI event mode 3                                    | N/A<br>0  | N/A<br>0  | N/A<br>0  | N/A<br>0  | N/A<br>0  | N/A<br>0  | COL9<br>0 | COL8<br>0 |
| 0x23    | GPIO_DIR1                               | GPIO data direction<br>0: input<br>1: output        | R7DD<br>0 | R6DD<br>0 | R5DD<br>0 | R4DD<br>0 | R3DD<br>0 | R2DD<br>0 | R1DD<br>0 | R0DD<br>0 |
| 0x24    | GPIO_DIR2                               | GPIO data direction<br>0: input<br>1: output        | C7DD<br>0 | C6DD<br>0 | C5DD<br>0 | C4DD<br>0 | C3DD<br>0 | C2DD<br>0 | C1DD<br>0 | C0DD<br>0 |
| 0x25    | GPIO_DIR3                               | GPIO data direction<br>0: input<br>1: output        | N/A<br>0  | N/A<br>0  | N/A<br>0  | N/A<br>0  | N/A<br>0  | N/A<br>0  | C9DD<br>0 | C8DD<br>0 |

**Table 9. Register Descriptions (continued)**

| ADDRESS | REGISTER NAME | REGISTER DESCRIPTION  | 7         | 6         | 5         | 4         | 3         | 2         | 1         | 0         |
|---------|---------------|---|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|
| 0x26    | GPIO_INT_LVL1 | GPIO edge/level detect<br>0: falling/low<br>1: rising/high        | R7IL<br>0 | R6IL<br>0 | R5IL<br>0 | R4IL<br>0 | R3IL<br>0 | R2IL<br>0 | R1IL<br>0 | R0IL<br>0 |
| 0x27    | GPIO_INT_LVL2 | GPIO edge/level detect<br>0: falling/low<br>1: rising/high        | C7IL<br>0 | C6IL<br>0 | C5IL<br>0 | C4IL<br>0 | C3IL<br>0 | C2IL<br>0 | C1IL<br>0 | C0IL<br>0 |
| 0x28    | GPIO_INT_LVL3 | GPIO edge/level detect<br>0: falling/low<br>1: rising/high        | N/A<br>0  | N/A<br>0  | N/A<br>0  | N/A<br>0  | N/A<br>0  | N/A<br>0  | C9IL<br>0 | C8IL<br>0 |
| 0x29    | DEBOUNCE_DIS1 | Debounce disable<br>0: debounce enabled<br>1: debounce disabled   | R7DD<br>0 | R6DD<br>0 | R5DD<br>0 | R4DD<br>0 | R3DD<br>0 | R2DD<br>0 | R1DD<br>0 | R0DD<br>0 |
| 0x2A    | DEBOUNCE_DIS2 | Debounce disable<br>0: debounce enabled<br>1: debounce disabled   | C7DD<br>0 | C6DD<br>0 | C5DD<br>0 | C4DD<br>0 | C3DD<br>0 | C2DD<br>0 | C1DD<br>0 | C0DD<br>0 |
| 0x2B    | DEBOUNCE_DIS3 | Debounce disable<br>0: debounce enabled<br>1: debounce disabled   | N/A<br>0  | N/A<br>0  | N/A<br>0  | N/A<br>0  | N/A<br>0  | N/A<br>0  | C9DD<br>0 | C8DD<br>0 |
| 0x2C    | GPIO_PULL1    | GPIO pull-up disable<br>0: pull-up enabled<br>1: pull-up disabled | R7PD<br>0 | R6PD<br>0 | R5PD<br>0 | R4PD<br>0 | R3PD<br>0 | R2PD<br>0 | R1PD<br>0 | R0PD<br>0 |
| 0x2D    | GPIO_PULL2    | GPIO pull-up disable<br>0: pull-up enabled<br>1: pull-up disabled | C7PD<br>0 | C6PD<br>0 | C5PD<br>0 | C4PD<br>0 | C3PD<br>0 | C2PD<br>0 | C1PD<br>0 | C0PD<br>0 |
| 0x2E    | GPIO_PULL3    | GPIO pull-up disable<br>0: pull-up enabled<br>1: pull-up disabled | N/A<br>0  | N/A<br>0  | N/A<br>0  | N/A<br>0  | N/A<br>0  | N/A<br>0  | C9PD<br>0 | C8PD<br>0 |
| 0x2F    | Reserved      |   |           |           |           |           |           |           |           |           |

**8.6.2.1 Configuration Register (Address 0x01)**

| BIT | NAME         | DESCRIPTION   |
|-----|--------------|---|
| 7   | AI           | Auto-increment for read and write operations; See below table for more information<br>0 = disabled<br>1 = enabled   |
| 6   | GPI_E_CFG    | GPI event mode configuration<br>0 = GPI events are tracked when keypad is locked<br>1 = GPI events are not tracked when keypad is locked  |
| 5   | OVR_FLOW_M   | Overflow mode<br>0 = disabled; Overflow data is lost<br>1 = enabled; Overflow data shifts with last event pushing first event out   |
| 4   | INT_CFG      | Interrupt configuration<br>0 = processor interrupt remains asserted (or low) if host tries to clear interrupt while there is still a pending key press, key release or GPI interrupt<br>1 = processor interrupt is deasserted for 50 $\mu$ s and reassert with pending interrupts |
| 3   | OVR_FLOW_IEN | Overflow interrupt enable<br>0 = disabled; $\overline{\text{INT}}$ is not asserted if the FIFO overflows<br>1 = enabled; $\overline{\text{INT}}$ becomes asserted if the FIFO overflows   |
| 2   | K_LCK_IEN    | Keypad lock interrupt enable<br>0 = disabled; $\overline{\text{INT}}$ is not asserted after a correct unlock key sequence<br>1 = enabled; $\overline{\text{INT}}$ becomes asserted after a correct unlock key sequence  |
| 1   | GPI_IEN      | GPI interrupt enable to host processor<br>0 = disabled; $\overline{\text{INT}}$ is not asserted for a change on a GPI<br>1 = enabled; $\overline{\text{INT}}$ becomes asserted for a change on a GPI  |
| 0   | KE_IEN       | Key events interrupt enable to host processor<br>0 = disabled; $\overline{\text{INT}}$ is not asserted when a key event occurs<br>1 = enabled; $\overline{\text{INT}}$ becomes asserted when a key event occurs   |

Bit 7 in this register is used to determine the programming mode. If it is low, all data bytes are written to the register defined by the command byte. If bit 7 is high, the value of the command byte is automatically incremented after each byte is written, and the next data byte is stored in the corresponding register. Registers are written in the sequence shown in [Table 9](#). Once the GPIO\_PULL3 register (0x2E) is written to, the command byte returns to register 0. Registers 0 and 2F are reserved and a command byte that references these registers is not acknowledged by the TCA8418.

The keypad lock interrupt enable determines if the interrupt pin is asserted when the key lock interrupt (see [Interrupt Status Register](#)) bit is set.

### 8.6.2.2 Interrupt Status Register, INT\_STAT (Address 0x02)

| BIT | NAME         | DESCRIPTION  |
|-----|--------------|--|
| 7   | N/A          | Always 0   |
| 6   | N/A          | Always 0   |
| 5   | N/A          | Always 0   |
| 4   | CAD_INT      | CTRL-ALT-DEL key sequence status. Requires writing a 1 to clear interrupts.<br>0 = interrupt not detected<br>1 = interrupt detected  |
| 3   | OVR_FLOW_INT | Overflow interrupt status. Requires writing a 1 to clear interrupts.<br>0 = interrupt not detected<br>1 = interrupt detected   |
| 2   | K_LCK_INT    | Keypad lock interrupt status. This is the interrupt to the processor when the keypad lock sequence is started. Requires writing a 1 to clear interrupts.<br>0 = interrupt not detected<br>1 = interrupt detected |
| 1   | GPI_INT      | GPI interrupt status. Requires writing a 1 to clear interrupts.<br>0 = interrupt not detected<br>1 = interrupt detected<br>Can be used to mask interrupts  |
| 0   | K_INT        | Key events interrupt status. Requires writing a 1 to clear interrupts.<br>0 = interrupt not detected<br>1 = interrupt detected   |

The INT\_STAT register is used to check which type of interrupt has been triggered. If the corresponding interrupt enable bits are set in the [Configuration Register](#), then a value of 1 in the corresponding bit will assert the  $\overline{\text{INT}}$  line low. An exception to this is the CAD\_INT bit, which will assert the CAD\_INT pin on YFP packages.

A read to this register will return which types of events have occurred. Writing a 1 to the bit will clear the interrupt, unless there is still data which has set the Interrupt (unread keys in the FIFO).

### 8.6.2.3 Key Lock and Event Counter Register, KEY\_LCK\_EC (Address 0x03)

| BIT | NAME                | DESCRIPTION   |
|-----|---------------------|---|
| 7   | N/A                 | Always 0  |
| 6   | K_LCK_EN            | Key lock enable<br>0 = disabled; Write a 0 to this bit to unlock the keypad manually<br>1 = enabled; Write a 1 to this bit to lock the keypad |
| 5   | LCK2                | Keypad lock status<br>0 = unlock (if LCK1 is 0 too)<br>1 = locked (if LCK1 is 1 too)  |
| 4   | LCK1                | Keypad lock status<br>0 = unlock (if LCK2 is 0 too)<br>1 = locked (if LCK2 is 1 too)  |
| 3   | KEC3 <sup>(1)</sup> | Key event count, Bit 3  |
| 2   | KEC2                | Key event count, Bit 2  |
| 1   | KEC1                | Key event count, Bit 1  |
| 0   | KEC0                | Key event count, Bit 0  |

(1) KEC[3:0] indicates how many key events are in the FIFO. For example, KEC[3:0] = 0b0000 = 0 events, KEC[3:0] = 0b0001 = 1 event and KEC[3:0] = 0b1010 = 10 events. As events happen (press or release), the count increases accordingly.

### 8.6.2.4 Key Event Registers (FIFO), KEY\_EVENT\_A–J (Address 0x04–0x0D)

| ADDRESS | REGISTER NAME <sup>(1)</sup> | REGISTER DESCRIPTION | BIT  |      |      |      |      |      |      |      |
|---------|------------------------------|----------------------|------|------|------|------|------|------|------|------|
|         |                              |                      | 7    | 6    | 5    | 4    | 3    | 2    | 1    | 0    |
| 0x04    | KEY_EVENT_A                  | Key event register A | KEA7 | KEA6 | KEA5 | KEA4 | KEA3 | KEA2 | KEA1 | KEA0 |

(1) Only KEY\_EVENT\_A register is shown

These registers – KEY\_EVENT\_A–J – function as a FIFO stack which can store up to 10 key presses and releases. The user first checks the INT\_STAT register to see if there are any interrupts. If so, then the Key Lock and Event Counter Register (KEY\_LCK\_EC, register 0x03) is read to see how many interrupts are stored. The INT\_STAT register is then read again to ensure no new events have come in. The KEY\_EVENT\_A register is then read as many times as there are interrupts. Each time a read happens, the count in the KEY\_LCK\_EC register reduces by 1. The data in the FIFO also moves down the stack by 1 too (from KEY\_EVENT\_J to KEY\_EVENT\_A). Once all events have been read, the key event count is at 0 and then KE\_INT bit can be cleared by writing a '1' to it.

In the KEY\_EVENT\_A register, KEA[6:0] indicates the key # pressed or released. A value of 0 to 80 indicate which key has been pressed or released in a keypad matrix. Values of 97 to 114 are for GPI events.

Bit 7 or KEA[7] indicate if a key press or key release has happened. A '0' means a key release happened. A '1' means a key has been pressed (which can be cleared on a read).

For example, 3 key presses and 3 key releases are stored as 6 words in the FIFO. As each word is read, the user knows if it is a key press or key release that occurred. Key presses such as CTRL+ALT+DEL are stored as 3 simultaneous key presses. Key presses and releases generate key event interrupts. The KE\_INT bit and /INT pin will not cleared until the FIFO is cleared of all events.

All registers can be read but for the purpose of the FIFO, the user should only read KEY\_EVENT\_A register. Once all the events in the FIFO have been read, reading of KEY\_EVENT\_A register will yield a zero value.

### 8.6.2.5 Keypad Lock1 to Lock2 Timer Register, KP\_LCK\_TIMER (Address 0x0E)

| ADDRESS | REGISTER NAME | REGISTER DESCRIPTION  | BIT |     |     |     |     |     |     |     |
|---------|---------------|---|-----|-----|-----|-----|-----|-----|-----|-----|
|         |               |   | 7   | 6   | 5   | 4   | 3   | 2   | 1   | 0   |
| 0x0E    | KP_LCK_TIMER  | Keypad lock interrupt mask timer and lock 1 to lock 2 timer | KL7 | KL6 | KL5 | KL4 | KL3 | KL2 | KL1 | KL0 |

KL[2:0] are for the Lock1 to Lock2 timer

KL[7:3] are for the interrupt mask timer

Lock1 to Lock2 timer must be non-zero for keylock to be enabled. The lock1 to lock2 bits ( KL[2:0] ) define the time in seconds the user has to press unlock key 2 after unlock key 1 before the key lock sequence times out. For more information, please see [Keypad Lock/Unlock](#).

If the keypad lock interrupt mask timer is non-zero, a key event interrupt (K\_INT) will be generated on any first key press. The second interrupt (K\_LCK\_IN) will only be generated when the correct unlock sequence has been completed. If either timer expires, the keylock state machine will reset.

When the interrupt mask timer is disabled ('0'), a key lock interrupt will trigger only when the correct unlock sequence is completed.

The interrupt mask timer should be set for the time it takes for the LCD to dim or turn off. For more information, please see [Keypad Lock Interrupt Mask Timer](#).

### 8.6.2.6 Unlock1 and Unlock2 Registers, UNLOCK1/2 (Address 0x0F-0x10)

| ADDRESS | REGISTER NAME | REGISTER DESCRIPTION | BIT   |       |       |       |       |       |       |       |
|---------|---------------|----------------------|-------|-------|-------|-------|-------|-------|-------|-------|
|         |               |                      | 7     | 6     | 5     | 4     | 3     | 2     | 1     | 0     |
| 0x0F    | Unlock1       | Unlock key 1         | UK1_7 | UK1_6 | UK1_5 | UK1_4 | UK1_3 | UK1_2 | UK1_1 | UK1_0 |
| 0x10    | Unlock2       | Unlock key 2         | UK2_7 | UK2_6 | UK2_5 | UK2_4 | UK2_3 | UK2_2 | UK2_1 | UK2_0 |

UK1[6:0] contains the key number used to unlock key 1

UK2[6:0] contains the key number used to unlock key 2

A '0' in either register disables the keylock function.

### 8.6.2.7 GPIO Interrupt Status Registers, GPIO\_INT\_STAT1–3 (Address 0x11–0x13)

These registers are used to check GPIO interrupt status. If the GPI\_INT bit is set in [INT\\_STAT](#) register, then the GPI which set the interrupt is marked with a 1 in the corresponding table. To clear the GPI\_INT bit, these registers must all be 0x00. A read to the register clears the bit.

| ADDRESS | REGISTER NAME  | REGISTER DESCRIPTION    | BIT  |      |      |      |      |      |      |      |
|---------|----------------|-------------------------|------|------|------|------|------|------|------|------|
|         |                |                         | 7    | 6    | 5    | 4    | 3    | 2    | 1    | 0    |
| 0x11    | GPIO_INT_STAT1 | GPIO Interrupt Status 1 | R7IS | R6IS | R5IS | R4IS | R3IS | R2IS | R1IS | R0IS |
| 0x12    | GPIO_INT_STAT2 | GPIO Interrupt Status 2 | C7IS | C6IS | C5IS | C4IS | C3IS | C2IS | C1IS | C0IS |
| 0x13    | GPIO_INT_STAT3 | GPIO Interrupt Status 3 | N/A  | N/A  | N/A  | N/A  | N/A  | N/A  | C9IS | C8IS |

### 8.6.2.8 GPIO Data Status Registers, GPIO\_DAT\_STAT1–3 (Address 0x14–0x16)

These registers show the GPIO state when read for inputs and outputs. Read these twice to clear them.

If debouncing is enabled, these registers return their default values until a change of state occurs at an input. Initial pin states can be read by disabling debouncing.

| ADDRESS | REGISTER NAME  | REGISTER DESCRIPTION | BIT  |      |      |      |      |      |      |      |
|---------|----------------|----------------------|------|------|------|------|------|------|------|------|
|         |                |                      | 7    | 6    | 5    | 4    | 3    | 2    | 1    | 0    |
| 0x14    | GPIO_DAT_STAT1 | GPIO Data Status 1   | R7DS | R6DS | R5DS | R4DS | R3DS | R2DS | R1DS | R0DS |
| 0x15    | GPIO_DAT_STAT2 | GPIO Data Status 2   | C7DS | C6DS | C5DS | C4DS | C3DS | C2DS | C1DS | C0DS |
| 0x16    | GPIO_DAT_STAT3 | GPIO Data Status 3   | N/A  | N/A  | N/A  | N/A  | N/A  | N/A  | C9DS | C8DS |

### 8.6.2.9 GPIO Data Out Registers, GPIO\_DAT\_OUT1–3 (Address 0x17–0x19)

These registers contain GPIO data to be written to GPIO out driver; inputs are not affected. This sets the output for the corresponding GPIO output.

| ADDRESS | REGISTER NAME | REGISTER DESCRIPTION | BIT  |      |      |      |      |      |      |      |
|---------|---------------|----------------------|------|------|------|------|------|------|------|------|
|         |               |                      | 7    | 6    | 5    | 4    | 3    | 2    | 1    | 0    |
| 0x17    | GPIO_DAT_OUT1 | GPIO Data Out 1      | R7DO | R6DO | R5DO | R4DO | R3DO | R2DO | R1DO | R0DO |
| 0x18    | GPIO_DAT_OUT2 | GPIO Data Out 2      | C7DO | C6DO | C5DO | C4DO | C3DO | C2DO | C1DO | C0DO |
| 0x19    | GPIO_DAT_OUT3 | GPIO Data Out 3      | N/A  | N/A  | N/A  | N/A  | N/A  | N/A  | C9DO | C8DO |

### 8.6.2.10 GPIO Interrupt Enable Registers, GPIO\_INT\_EN1–3 (Address 0x1A–0x1C)

These registers enable interrupts (bit value 1) or disable interrupts (bit value '0') for general purpose inputs (GPI) only. If the input changes on a pin which is setup as a GPI, then the GPI\_INT bit will be set in the [INT\\_STAT](#) register.

A bit value of '0' in any of the unreserved bits disables the corresponding pin's ability to generate an interrupt when the state of the input changes. This is the default value.

A bit value of 1 in any of the unreserved bits enables the corresponding pin's ability to generate an interrupt when the state of the input changes.

| ADDRESS | REGISTER NAME | REGISTER DESCRIPTION    | BIT  |      |      |      |      |      |      |      |
|---------|---------------|-------------------------|------|------|------|------|------|------|------|------|
|         |               |                         | 7    | 6    | 5    | 4    | 3    | 2    | 1    | 0    |
| 0x1A    | GPIO_INT_EN1  | GPIO Interrupt Enable 1 | R7IE | R6IE | R5IE | R4IE | R3IE | R2IE | R1IE | R0IE |
| 0x1B    | GPIO_INT_EN2  | GPIO Interrupt Enable 2 | C7IE | C6IE | C5IE | C4IE | C3IE | C2IE | C1IE | C0IE |
| 0x1C    | GPIO_INT_EN3  | GPIO Interrupt Enable 3 | N/A  | N/A  | N/A  | N/A  | N/A  | N/A  | C9IE | C8IE |

### 8.6.2.11 Keypad or GPIO Selection Registers, KP\_GPIO1–3 (Address 0x1D–0x1F)

A bit value of '0' in any of the unreserved bits puts the corresponding pin in GPIO mode. A pin in GPIO mode can be configured as an input or an output in the [GPIO\\_DIR1-3](#) registers. This is the default value.

A 1 in any of these bits puts the pin in key scan mode and becomes part of the keypad array, then it is configured as a row or column accordingly (this is not adjustable).

| ADDRESS | REGISTER NAME | REGISTER DESCRIPTION | BIT  |      |      |      |      |      |      |      |
|---------|---------------|----------------------|------|------|------|------|------|------|------|------|
|         |               |                      | 7    | 6    | 5    | 4    | 3    | 2    | 1    | 0    |
| 0x1D    | KP_GPIO1      | Keypad/GPIO Select 1 | ROW7 | ROW6 | ROW5 | ROW4 | ROW3 | ROW2 | ROW1 | ROW0 |
| 0x1E    | KP_GPIO2      | Keypad/GPIO Select 2 | COL7 | COL6 | COL5 | COL4 | COL3 | COL2 | COL1 | COL0 |
| 0x1F    | KP_GPIO3      | Keypad/GPIO Select 3 | N/A  | N/A  | N/A  | N/A  | N/A  | N/A  | COL9 | COL8 |

### 8.6.2.12 GPI Event Mode Registers, GPI\_EM1–3 (Address 0x20–0x22)

A bit value of '0' in any of the unreserved bits indicates that it is not part of the event FIFO. This is the default value.

A 1 in any of these bits means it is part of the event FIFO. When a pin is setup as a GPI and has a value of 1 in the Event Mode register, then any key presses will be added to the FIFO. Please see [Key Event Table](#) for more information.

| ADDRESS | REGISTER NAME | REGISTER DESCRIPTION    | BIT  |      |      |      |      |      |      |      |
|---------|---------------|-------------------------|------|------|------|------|------|------|------|------|
|         |               |                         | 7    | 6    | 5    | 4    | 3    | 2    | 1    | 0    |
| 0x20    | GPI_EM1       | GPI Event Mode Select 1 | ROW7 | ROW6 | ROW5 | ROW4 | ROW3 | ROW2 | ROW1 | ROW0 |
| 0x21    | GPI_EM2       | GPI Event Mode Select 2 | COL7 | COL6 | COL5 | COL4 | COL3 | COL2 | COL1 | COL0 |
| 0x23    | GPI_EM3       | GPI Event Mode Select 3 | N/A  | N/A  | N/A  | N/A  | N/A  | N/A  | COL9 | COL8 |

### 8.6.2.13 GPIO Data Direction Registers, GPIO\_DIR1–3 (Address 0x23–0x25)

A bit value of '0' in any of the unreserved bits sets the corresponding pin as an input. This is the default value.

A 1 in any of these bits sets the pin as an output.

| ADDRESS | REGISTER NAME | REGISTER DESCRIPTION | BIT  |      |      |      |      |      |      |      |
|---------|---------------|----------------------|------|------|------|------|------|------|------|------|
|         |               |                      | 7    | 6    | 5    | 4    | 3    | 2    | 1    | 0    |
| 0x23    | GPIO_DIR1     | GPIO Direction 1     | R7DD | R6DD | R5DD | R4DD | R3DD | R2DD | R1DD | R0DD |
| 0x24    | GPIO_DIR2     | GPIO Direction 2     | C7DD | C6DD | C5DD | C4DD | C3DD | C2DD | C1DD | C0DD |
| 0x25    | GPIO_DIR3     | GPIO Direction 3     | N/A  | N/A  | N/A  | N/A  | N/A  | N/A  | C9DD | C8DD |

**8.6.2.14 GPIO Edge/Level Detect Registers, GPIO\_INT\_LVL1–3 (Address 0x26–0x28)**

A bit value of '0' indicates that interrupt will be triggered on a high-to-low/low-level transition for the inputs in GPIO mode. This is the default value.

A bit value of 1 indicates that interrupt will be triggered on a low-to-high/high-level value for the inputs in GPIO mode.

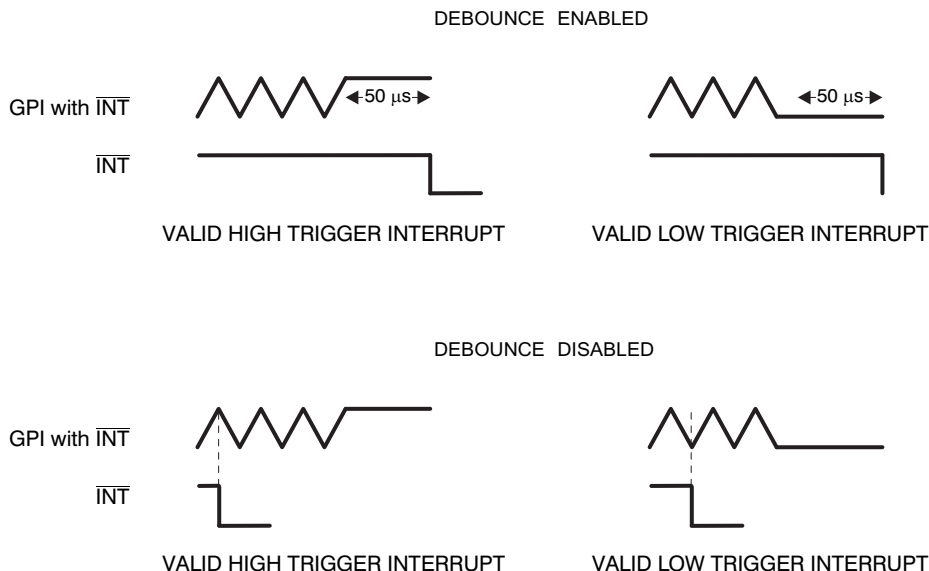
| ADDRESS | REGISTER NAME | REGISTER DESCRIPTION     | BIT  |      |      |      |      |      |      |      |
|---------|---------------|--------------------------|------|------|------|------|------|------|------|------|
|         |               |                          | 7    | 6    | 5    | 4    | 3    | 2    | 1    | 0    |
| 0x26    | GPIO_INT_LVL1 | GPIO Edge/Level Detect 1 | R7IL | R6IL | R5IL | R4IL | R3IL | R2IL | R1IL | R0IL |
| 0x27    | GPIO_INT_LVL2 | GPIO Edge/Level Detect 2 | C7IL | C6IL | C5IL | C4IL | C3IL | C2IL | C1IL | C0IL |
| 0x28    | GPIO_INT_LVL3 | GPIO Edge/Level Detect 3 | N/A  | N/A  | N/A  | N/A  | N/A  | N/A  | C9IL | C8IL |

**8.6.2.15 Debounce Disable Registers, DEBOUNCE\_DIS1–3 (Address 0x29–0x2B)**

This is for pins configured as inputs. A bit value of '0' in any of the unreserved bits enables the debounce. This is the default value

A bit value of '1' disables the debounce.

| ADDRESS | REGISTER NAME | REGISTER DESCRIPTION | BIT  |      |      |      |      |      |      |      |
|---------|---------------|----------------------|------|------|------|------|------|------|------|------|
|         |               |                      | 7    | 6    | 5    | 4    | 3    | 2    | 1    | 0    |
| 0x29    | DEBOUNCE_DIS1 | Debounce Disable 1   | R7DD | R6DD | R5DD | R4DD | R3DD | R2DD | R1DD | R0DD |
| 0x30    | DEBOUNCE_DIS2 | Debounce Disable 2   | C7DD | C6DD | C5DD | C4DD | C3DD | C2DD | C1DD | C0DD |
| 0x2B    | DEBOUNCE_DIS3 | Debounce Disable 3   | N/A  | N/A  | N/A  | N/A  | N/A  | N/A  | C9DD | C8DD |



Debounce disable will have the same effect for GPI mode or for rows in keypad scanning mode. The  $\overline{\text{RESET}}$  input always has a 50- $\mu\text{s}$  debounce time.

The debounce time for inputs is the time required for the input to be stable to be noticed. This time is 50  $\mu\text{s}$ .

The debounce time for the keypad is for the columns only. The minimum time is 25 ms. All columns are scanned once every 25 ms to detect any key presses. Two full scans are required to see if any keys were pressed. If the first scan is done just after a key press, it takes 25 ms to detect the key press. If the first scan is done much later than the key press, it will take 40 ms to detect a key press.

**8.6.2.16 GPIO pull-up Disable Register, GPIO\_PULL1–3 (Address 0x2C–0x2E)**

This register enables or disables pull-up registers from inputs.

A bit value of '0' will enable the internal pull-up resistors. This is the default value.

A bit value of 1 will disable the internal pull-up resistors.

| ADDRESS | REGISTER NAME | REGISTER DESCRIPTION   | BIT  |      |      |      |      |      |      |      |
|---------|---------------|------------------------|------|------|------|------|------|------|------|------|
|         |               |                        | 7    | 6    | 5    | 4    | 3    | 2    | 1    | 0    |
| 0x2C    | GPIO_PULL1    | GPIO pull-up Disable 1 | R7PD | R6PD | R5PD | R4PD | R3PD | R2PD | R1PD | R0PD |
| 0x3D    | GPIO_PULL2    | GPIO pull-up Disable 2 | C7PD | C6PD | C5PD | C4PD | C3PD | C2PD | C1PD | C0PD |
| 0x2E    | GPIO_PULL3    | GPIO pull-up Disable 3 | N/A  | N/A  | N/A  | N/A  | N/A  | N/A  | C9PD | C8PD |

### 8.6.3 CAD Interrupt Errata

#### 8.6.3.1 Description

In the Interrupt Status Register (see [Table 10](#)), bit 4 is used to indicate the detection of a CTRL-ALT-DEL key sequence. Certain key press sequences trigger this bit to register a CAD\_INT improperly.

**Table 10. Interrupt Status Register, INT\_STAT (Address 0x02)**

| BIT | NAME         | DESCRIPTION  |
|-----|--------------|--|
| 7   | N/A          | Always 0   |
| 6   | N/A          | Always 0   |
| 5   | N/A          | Always 0   |
| 4   | CAD_INT      | CTRL-ALT-DEL key sequence status. Requires writing a 1 to clear interrupts.<br>0 = interrupt not detected<br>1 = interrupt detected  |
| 3   | OVR_FLOW_INT | Overflow interrupt status. Requires writing a 1 to clear interrupts.<br>0 = interrupt not detected<br>1 = interrupt detected   |
| 2   | K_LCK_INT    | Keypad lock interrupt status. This is the interrupt to the processor when the keypad lock sequence is started. Requires writing a 1 to clear interrupts.<br>0 = interrupt not detected<br>1 = interrupt detected |
| 1   | GPI_INT      | GPI interrupt status. Requires writing a 1 to clear interrupts.<br>0 = interrupt not detected<br>1 = interrupt detected<br>Can be used to mask interrupts  |
| 0   | K_INT        | Key events interrupt status. Requires writing a 1 to clear interrupts.<br>0 = interrupt not detected<br>1 = interrupt detected   |

The following key press combinations will cause a false CAD\_INT:

- 1 + 11
- 1 + 21
- 21 + 1 + 11

#### 8.6.3.2 System Impact

This device has an individual pin for the CAD\_INT unlike the TCA8418. The CAD\_INT pin falsely flag the processor in addition to the Interrupt Status Register's CAD\_INT bit being High when a CAD sequence did not occur.

#### 8.6.3.3 System Workaround

There is no system workaround to avoid the Interrupt Status Register to indicate a CAD\_INT sequence being detected.

## 8.6.4 Overflow Errata

### 8.6.4.1 Description

The TCA8418 allows for overflow detection of the 10 byte FIFO of key-press and release events. For overflow to be enabled, both Bit\_3 and Bit\_5 of the Configuration Register (see [Table 11](#)) must be set High. If only Bit\_3 set high, no overflow interrupt is generated.

**Table 11. Configuration Register (Address 0x01)**

| BIT | NAME         | DESCRIPTION   |
|-----|--------------|---|
| 7   | AI           | Auto-increment for read and write operation<br>0 = disabled<br>1 = enabled  |
| 6   | CPI_E_CFG    | GPI even mode configuration<br>0 = GPI events are tracked when keypad is locked<br>1 = GPI events are not tracked when keypad is locked   |
| 5   | OVR_FLOW_M   | Overflow mode<br>0 = disabled; overflow data is lost<br>1 = enabled.  |
| 4   | INT_CFG      | Overflow data shifts with last event pushing first event out interrupt configuration.<br>0 = processor interrupt remains asserted (or low) if host tries to clear interrupt while there is still a pending key press, key release or GPI interrupt<br>1 = processor interrupt is deasserted for 50 $\mu$ s and reassert with pending interrupts |
| 3   | OVR_FLOW_IEN | Overflow interrupt enable<br>0 = disabled<br>1 = enabled  |
| 2   | K_LCK_IEN    | Keypad lock interrupt enable<br>0 = disabled<br>1 = enabled   |

### 8.6.4.2 System Impact

Enabling the Overflow improperly may lead to data loss.

### 8.6.4.3 System Workaround

The FIFO stack should be read as new information is passed in. Since the Overflow interrupt is triggered once that stack has begun to overflow, meaning data is already being lost.

## 9 Application and Implementation

### NOTE

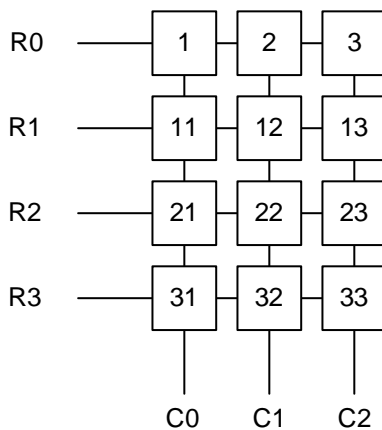
Information in the following applications sections is not part of the TI component specification, and TI does not warrant its accuracy or completeness. TI's customers are responsible for determining suitability of components for their purposes. Customers should validate and test their design implementation to confirm system functionality.

### 9.1 Application Information

#### 9.1.1 Ghosting Considerations

The TCA8418 supports multiple key presses accurately. Applications requiring three-key combinations (such as <Ctrl><Alt><Del>, or any other combinations) must ensure that the three keys are wired in appropriate key positions to avoid ghosting (or appearing like a 4th key has been pressed).

To avoid ghosting, it is best to keep 3-button combinations that are pressed on separate rows and columns. Consider the situation with the keypad described in [Figure 26](#).



**Figure 26. Example Keypad**

In the keypad setup in [Figure 26](#), there is a 4x3 keypad matrix, connected to ROW0-ROW3, and COL0-COL2. All of the ROWs are configured as inputs with pullup resistors. The COLs are configured as outputs, driving low. When a key press is made, one of the ROW inputs will be pulled low, letting the TCA8418 know that a key has been pressed, and the TCA8418 will then start the key scanning algorithm. During this algorithm, it sweeps the output low across the columns, such that only 1 column is driven low at a time. While this is done to each column, the TCA8418 will read the ROW inputs, to determine which keys on a column are being pressed.

Ghosting can occur when multiple keys are pressed that can make it appear that additional keys (which are not being pressed) are being pressed.

Application Information (continued)

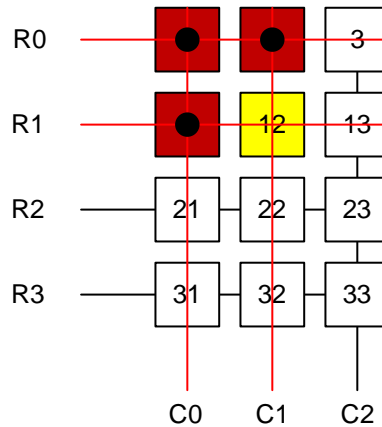


Figure 27. Incorrect 3 Button Combination

In [Figure 27](#), keys 1, 2, and 11 are pressed, which causes a ghosting issue. Since R1 becomes pulled to ground through key 1 (which is pulled through key 2 when C1 is transmitting a low), when C1 is driving low, the TCA8418 will see a low signal at both R0 and R1. This will falsely trigger key 12 as being pressed (the key highlighted as yellow).

The reason for this is that keypad matrices will short the columns to the rows connected together. When C1 is driving low, the low gets transmitted onto R0 via key 2. Key 1 is being pressed, which also shorts C0 to ground. Key 11 is pressed, which then shorts R1 to C0. In this process, R1 is shorted to C1, which is the reason ghosting occurs.

Keypad matrices can support multiple key presses properly, if care is taken when choosing the layout. In [Figure 28](#), we see a 3 button combination which work as expected. Keys 1, 11, and 21 are pressed (this also is the combination that will set the <Ctrl><Alt><Del> interrupt, see [Control-Alt-Delete Support](#) for more information).

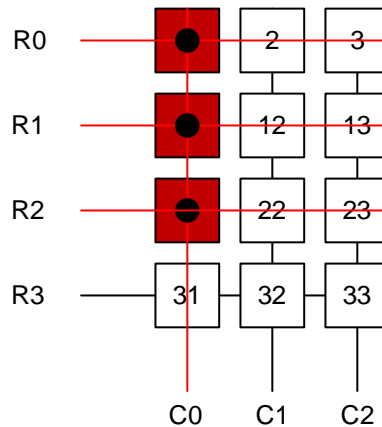
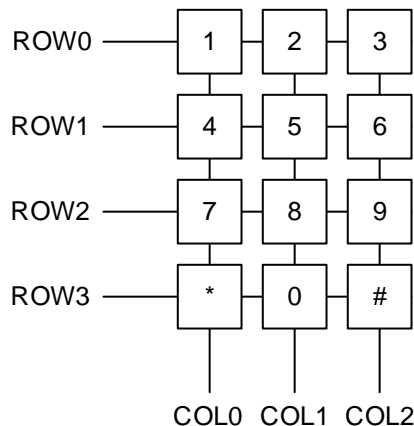


Figure 28. Correct 3 Button Combination

## 9.2 Typical Application

Figure 29 shows a typical application of the TCA8418. In this specific example, a common 12 key number pad layout is used. This number pad has keys for numbers 0 to 9, \*, and #.



**Figure 29. Typical Application**

### 9.2.1 Design Requirements

The system designer needs to know a few key pieces in order to design their system for the TCA8418.

- The number of keys desired
- Whether the keys will be multiplexed or not
- The layout of the multiplexed keys
- Unused keys be tied to VCC through a pullup resistor (10 kΩ)

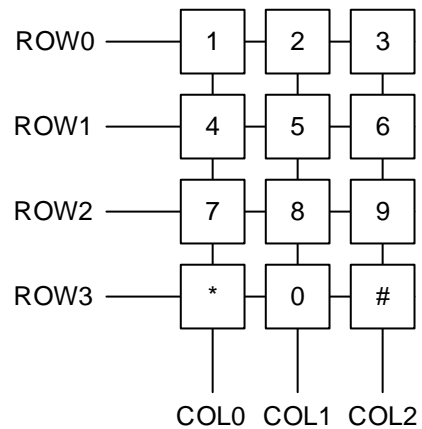
### 9.2.2 Detailed Design Procedure

#### 9.2.2.1 Designing the Hardware Layout

The first steps towards designing a keypad array is to determine the desired layout, and to map each key to the appropriate value which will show up in the FIFO. For this example, the number pad below is the physical location of the keys that are desired. The layout is a 4 x 3 array, using rows 0-3 and columns 0-2. For this example, we will not assume any of the other pins will be used.

The following behavior is desired for this example design

- All keys in the keypad array to be added to the FIFO upon a key press
- Attempting to clear the interrupt before the proper registers have been cleared to de-assert the  $\overline{\text{INT}}$  pin for 50  $\mu\text{s}$ , then assert the  $\text{INT}$  pin.
- No additional pins are being used, other than the keypad array
- Keypad lock support, requiring that the unlock combination be '#, 1' which must be pressed within 2 seconds of each other
- Keypad lock interrupt mask timer of 10 seconds to match the back light auto-turn off with 10 seconds of no interrupt
- Hardware debouncing to be enabled

**Typical Application (continued)**

**Figure 30. Example Keypad**

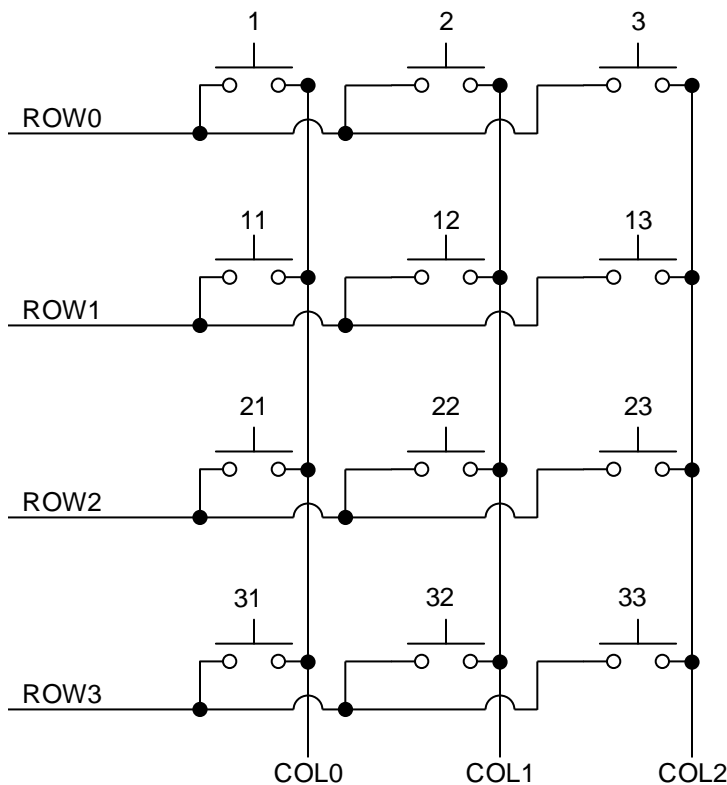
Since the TCA8418 reports keys pressed according to the values in the key value table, it is important to know the TCA8418 values for the key locations.

According to the key event table, the key presses are assigned in [Table 12](#).

**Table 12. Key Press Assignment**

| Keypad Button                   | 1 | 2 | 3 | 4  | 5  | 6  | 7  | 8  | 9  | *  | 0  | #  |
|---------------------------------|---|---|---|----|----|----|----|----|----|----|----|----|
| Key Event Table Value (Decimal) | 1 | 2 | 3 | 11 | 12 | 13 | 21 | 22 | 23 | 31 | 32 | 33 |

The schematic for this keypad layout is shown in *figure (schematic below)* with the key event table values. Note that no external pullup resistors are needed, because the TCA8418 has integrated pullup resistors.



**Figure 31. Keypad Schematic**

### 9.2.2.2 Configuring the Registers

The next step to design a keypad array for the TCA8418 is to configure the appropriate hardware registers.

The registers that must be modified for the desired features are the following

| STEP                         | REGISTER TO EDIT    | VALUE TO WRITE | DESCRIPTION   |
|------------------------------|---------------------|----------------|---|
| Setup keypad array           | KP_GPIO1 (0x1D)     | 0x0F           | Set ROW0-ROW3 to KP Matrix  |
|                              | KP_GPIO2 (0x1E)     | 0x07           | Set COL0-COL2 to KP Matrix  |
|                              | KP_GPIO3 (0x1F)     | 0x00           | Set COL8-COL9 to GPIO   |
| Setup Interrupts             | CFG (0x01)          | 0x95           | Set the KE_IEN, K_LCK_IEN, INT_CFG, and AI bits                         |
| Setup Unlock Key Combination | UNLOCK1 (0x0F)      | 0x21           | Set first unlock key to key 33  |
|                              | UNLOCK2 (0x10)      | 0x01           | Set second unlock key to key 1  |
| Set Keypad Lock Timers       | KP_LCK_TIMER (0x0E) | 0x52           | Lock1 to Lock2 set to 2 seconds. Interrupt mask timer set to 10 seconds |

### 9.2.3 Application Curves

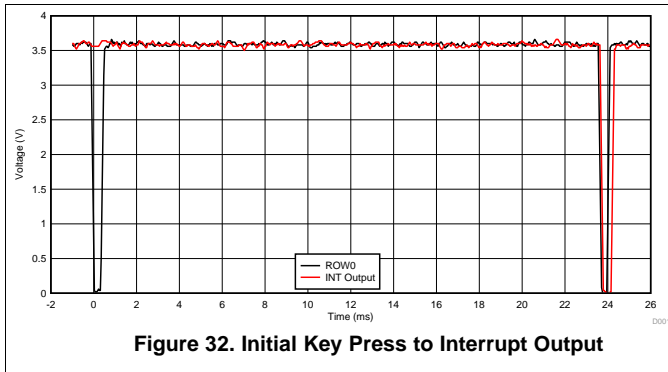


Figure 32. Initial Key Press to Interrupt Output

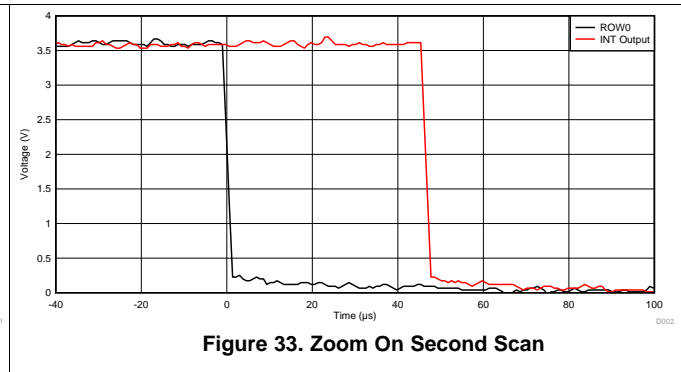


Figure 33. Zoom On Second Scan

## 10 Power Supply Recommendations

In the event of a glitch or data corruption, TCA8418 can be reset to its default conditions by using the power-on reset feature. Power-on reset requires that the device go through a power cycle to be completely reset. This reset also happens when the device is powered on for the first time in an application.

The two types of power-on reset are shown in [Figure 34](#) and [Figure 35](#).

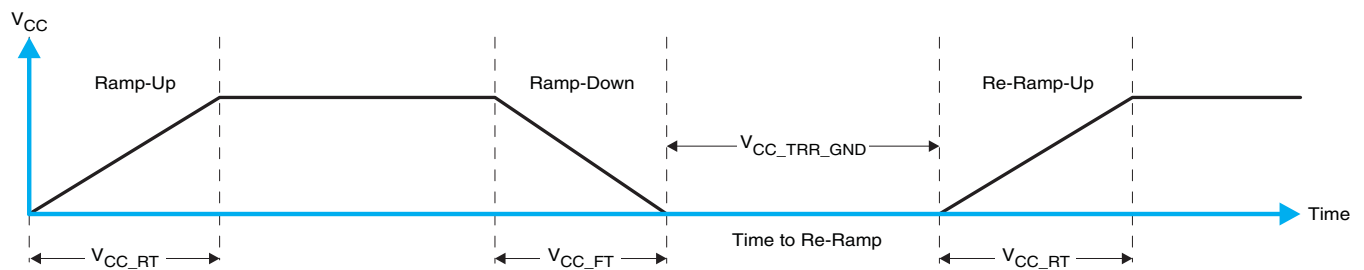


Figure 34.  $V_{CC}$  is Lowered Below 0.2 V or 0 V and Then Ramped Up to  $V_{CC}$

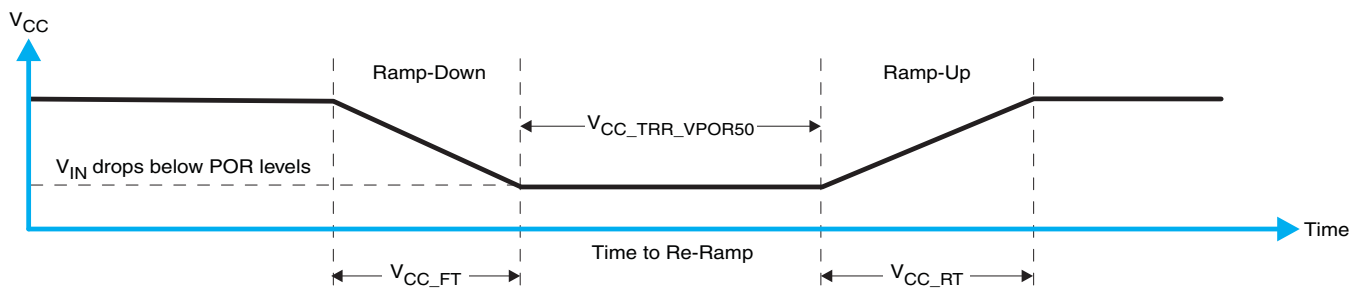


Figure 35.  $V_{CC}$  is Lowered Below the POR Threshold, Then Ramped Back Up to  $V_{CC}$

[Table 13](#) specifies the performance of the power-on reset feature for TCA8418 for both types of power-on reset.

**Table 13. Recommended Supply Sequencing and Ramp Rates<sup>(1)</sup>**

| PARAMETER            |   |                               | MIN   | TYP | MAX | UNIT |
|----------------------|---|-------------------------------|-------|-----|-----|------|
| $V_{CC\_FT}$         | Fall rate   | See <a href="#">Figure 34</a> | 1     |     | 100 | ms   |
| $V_{CC\_RT}$         | Rise rate   | See <a href="#">Figure 34</a> | 0.01  |     | 100 | ms   |
| $V_{CC\_TRR\_GND}$   | Time to re-ramp (when $V_{CC}$ drops to GND)                    | See <a href="#">Figure 34</a> | 0.001 |     |     | ms   |
| $V_{CC\_TRR\_POR50}$ | Time to re-ramp (when $V_{CC}$ drops to $V_{POR\_MIN} - 50$ mV) | See <a href="#">Figure 35</a> | 0.001 |     |     | ms   |

(1)  $T_A = -40^{\circ}\text{C}$  to  $85^{\circ}\text{C}$  (unless otherwise noted)

**Table 13. Recommended Supply Sequencing and Ramp Rates<sup>0</sup> (continued)**

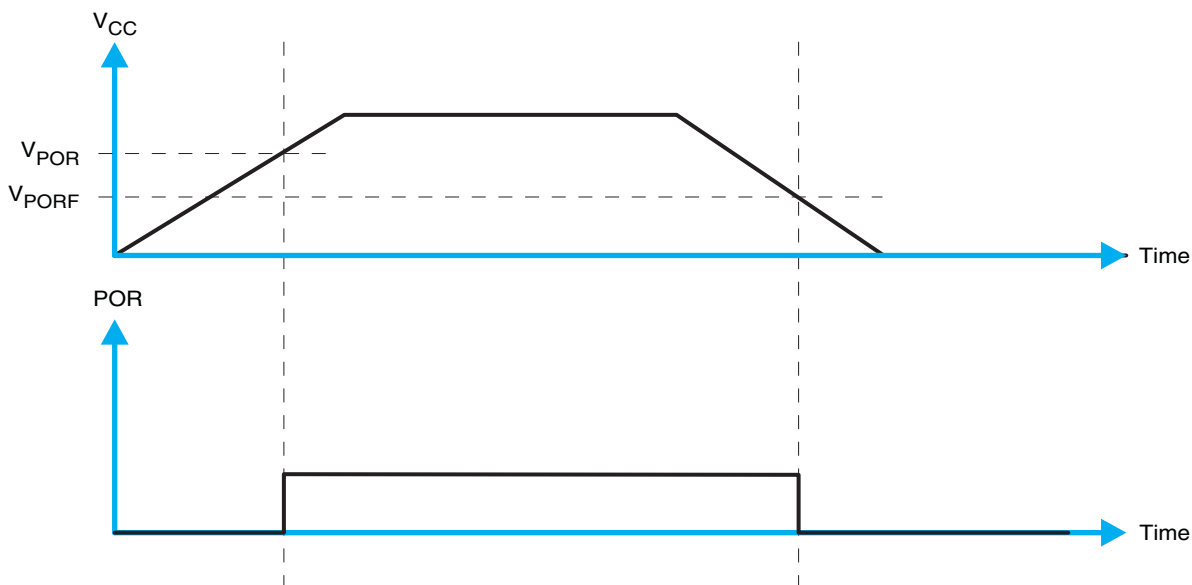
| PARAMETER    |  | MIN  | TYP | MAX  | UNIT    |
|--------------|--|------|-----|------|---------|
| $V_{CC\_GH}$ | Level that $V_{CC}$ can glitch down to, but not cause a functional disruption when $V_{CCX\_GW} = 1 \mu s$ |      |     | 1.2  | V       |
| $V_{CC\_GW}$ | Glitch width that will not cause a functional disruption when $V_{CCX\_GH} = 0.5 \times V_{CCx}$           |      |     | 10   | $\mu s$ |
| $V_{PORF}$   | Voltage trip point of POR on falling $V_{CC}$  | 0.76 |     | 1.15 | V       |
| $V_{PORR}$   | Voltage trip point of POR on rising $V_{CC}$   | 1.03 |     | 1.43 | V       |

Glitches in the power supply can also affect the power-on reset performance of this device. The glitch width ( $V_{CC\_GW}$ ) and height ( $V_{CC\_GH}$ ) are dependent on each other. The bypass capacitance, source impedance, and device impedance are factors that affect power-on reset performance. Figure 36 and Table 13 provide more information on how to measure these specifications.



**Figure 36. Glitch Width and Glitch Height**

$V_{POR}$  is critical to the power-on reset.  $V_{POR}$  is the voltage level at which the reset condition is released and all the registers and the I<sup>2</sup>C/SMBus state machine are initialized to their default states. The value of  $V_{POR}$  differs based on the  $V_{CC}$  being lowered to or from 0. Figure 37 and Table 13 provide more details on this specification.



**Figure 37.  $V_{POR}$**

For proper operation of the power-on reset feature, use as directed in the previous figures and table above.

## 11 Layout

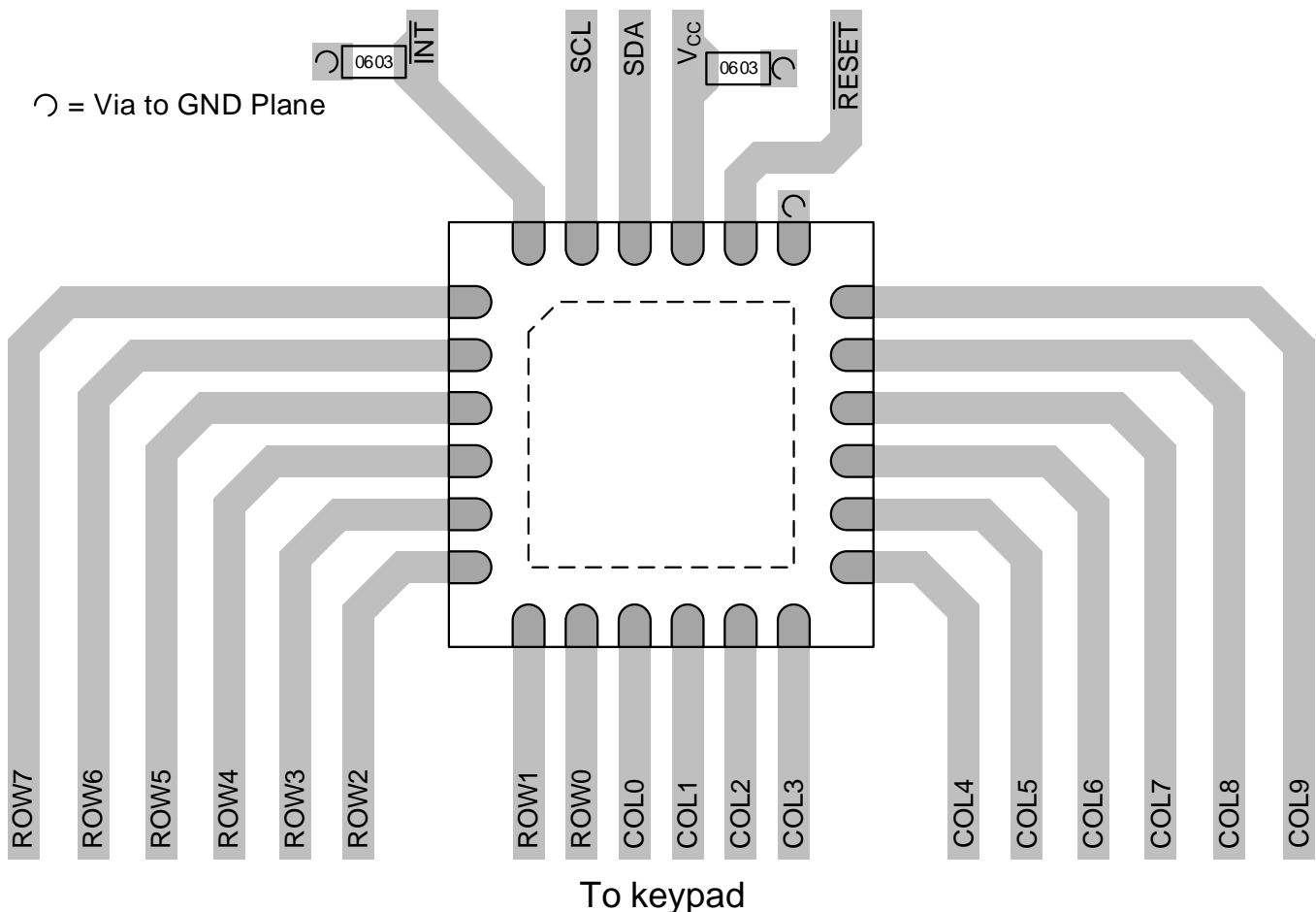
### 11.1 Layout Guidelines

For printed circuit board (PCB) layout of the TCA8418, common PCB layout practices should be followed, but additional concerns related to high-speed data transfer, such as matched impedances and differential pairs are not a concern for I<sup>2</sup>C signal speeds.

In all PCB layouts, it is best practice to avoid right angles in signal traces, to fan out signal traces away from each other upon leaving the vicinity of an integrated circuit (IC), and to use thicker trace widths to carry higher amounts of current that commonly pass through power and ground traces. Bypass and de-coupling capacitors are commonly used to control the voltage on the VCC pin, using a larger capacitor to provide additional power in the event of a short power supply glitch and a smaller capacitor to filter out high-frequency ripple. These capacitors should be placed as close to the TCA8418 as possible.

For the layout example provided in [Layout Example](#), a 4 layer board is required to route all of the signals. The layout example shows a way to route the signals out from the device, which can eventually be brought up to the top layer (or any required layer) with the use of a via. This technique is not demonstrated in this example due to the complexity of the layout.

### 11.2 Layout Example



**Figure 38. RTW Package Layout Example**

## 12 器件和文档支持

### 12.1 接收文档更新通知

要接收文档更新通知，请导航至 [TI.com.cn](http://TI.com.cn) 上的器件产品文件夹。单击右上角的 [通知我](#) 进行注册，即可每周接收产品信息更改摘要。有关更改的详细信息，请查阅已修订文档中包含的修订历史记录

### 12.2 社区资源

下列链接提供到 TI 社区资源的连接。链接的内容由各个分销商“按照原样”提供。这些内容并不构成 TI 技术规范，并且不一定反映 TI 的观点；请参阅 TI 的 [《使用条款》](#)。

**TI E2E™ 在线社区** [TI 的工程师对工程师 \(E2E\) 社区](#)。此社区的创建目的在于促进工程师之间的协作。在 [e2e.ti.com](http://e2e.ti.com) 中，您可以咨询问题、分享知识、拓展思路并与同行工程师一道帮助解决问题。

**设计支持** [TI 参考设计支持](#) 可帮助您快速查找有帮助的 E2E 论坛、设计支持工具以及技术支持的联系信息。

### 12.3 商标

E2E is a trademark of Texas Instruments.  
All other trademarks are the property of their respective owners.

### 12.4 静电放电警告



这些装置包含有限的内置 ESD 保护。存储或装卸时，应将导线一起截短或将装置放置于导电泡棉中，以防止 MOS 门极遭受静电损伤。

### 12.5 术语表

[SLYZ022](#) — *TI 术语表*。

这份术语表列出并解释术语、缩写和定义。

## 13 机械、封装和可订购信息

以下页面包含机械、封装和可订购信息。这些信息是指定器件的最新可用数据。数据如有变更，恕不另行通知，且不会对此文档进行修订。如需获取此产品说明书的浏览器版本，请参阅左侧的导航栏。

**PACKAGING INFORMATION**

| Orderable part number       | Status<br>(1) | Material type<br>(2) | Package   Pins  | Package qty   Carrier | RoHS<br>(3) | Lead finish/<br>Ball material<br>(4) | MSL rating/<br>Peak reflow<br>(5) | Op temp (°C) | Part marking<br>(6) |
|-----------------------------|---------------|----------------------|-----------------|-----------------------|-------------|--------------------------------------|-----------------------------------|--------------|---------------------|
| <a href="#">TCA8418RTWR</a> | Active        | Production           | WQFN (RTW)   24 | 3000   LARGE T&R      | Yes         | NIPDAU                               | Level-2-260C-1 YEAR               | -40 to 85    | PZ418               |
| TCA8418RTWR.A               | Active        | Production           | WQFN (RTW)   24 | 3000   LARGE T&R      | Yes         | NIPDAU                               | Level-2-260C-1 YEAR               | -40 to 85    | PZ418               |
| TCA8418RTWRG4               | Active        | Production           | WQFN (RTW)   24 | 3000   LARGE T&R      | Yes         | NIPDAU                               | Level-2-260C-1 YEAR               | -40 to 85    | PZ418               |
| TCA8418RTWRG4.A             | Active        | Production           | WQFN (RTW)   24 | 3000   LARGE T&R      | Yes         | NIPDAU                               | Level-2-260C-1 YEAR               | -40 to 85    | PZ418               |

(1) **Status:** For more details on status, see our [product life cycle](#).

(2) **Material type:** When designated, preproduction parts are prototypes/experimental devices, and are not yet approved or released for full production. Testing and final process, including without limitation quality assurance, reliability performance testing, and/or process qualification, may not yet be complete, and this item is subject to further changes or possible discontinuation. If available for ordering, purchases will be subject to an additional waiver at checkout, and are intended for early internal evaluation purposes only. These items are sold without warranties of any kind.

(3) **RoHS values:** Yes, No, RoHS Exempt. See the [TI RoHS Statement](#) for additional information and value definition.

(4) **Lead finish/Ball material:** Parts may have multiple material finish options. Finish options are separated by a vertical ruled line. Lead finish/Ball material values may wrap to two lines if the finish value exceeds the maximum column width.

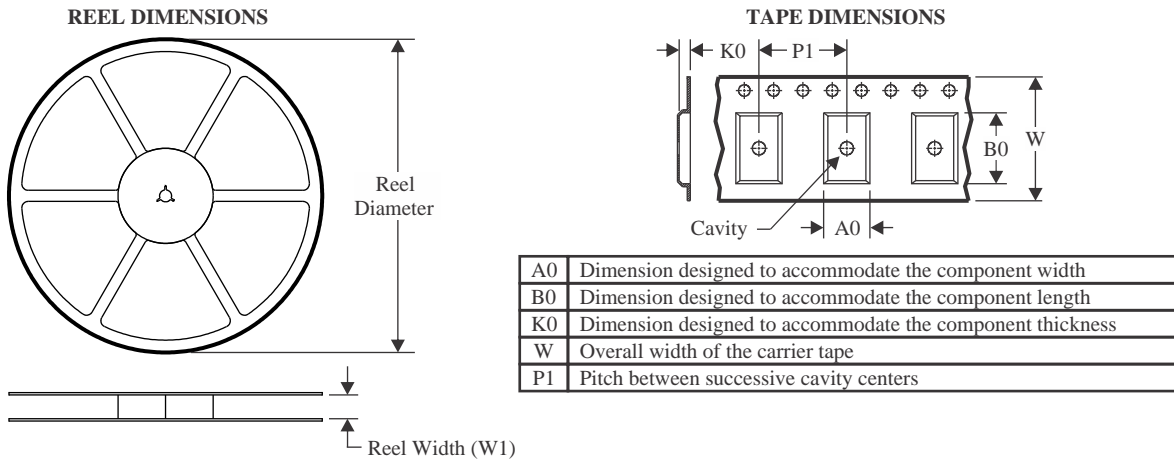
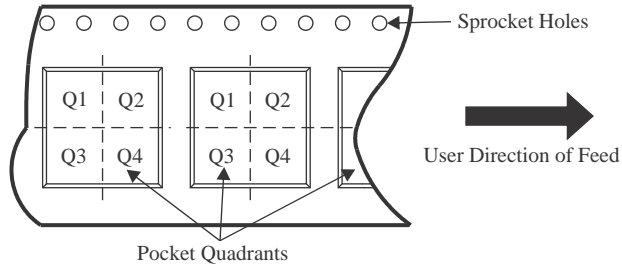
(5) **MSL rating/Peak reflow:** The moisture sensitivity level ratings and peak solder (reflow) temperatures. In the event that a part has multiple moisture sensitivity ratings, only the lowest level per JEDEC standards is shown. Refer to the shipping label for the actual reflow temperature that will be used to mount the part to the printed circuit board.

(6) **Part marking:** There may be an additional marking, which relates to the logo, the lot trace code information, or the environmental category of the part.

Multiple part markings will be inside parentheses. Only one part marking contained in parentheses and separated by a "~" will appear on a part. If a line is indented then it is a continuation of the previous line and the two combined represent the entire part marking for that device.

**Important Information and Disclaimer:** The information provided on this page represents TI's knowledge and belief as of the date that it is provided. TI bases its knowledge and belief on information provided by third parties, and makes no representation or warranty as to the accuracy of such information. Efforts are underway to better integrate information from third parties. TI has taken and continues to take reasonable steps to provide representative and accurate information but may not have conducted destructive testing or chemical analysis on incoming materials and chemicals. TI and TI suppliers consider certain information to be proprietary, and thus CAS numbers and other limited information may not be available for release.

In no event shall TI's liability arising out of such information exceed the total purchase price of the TI part(s) at issue in this document sold by TI to Customer on an annual basis.

**TAPE AND REEL INFORMATION**

**QUADRANT ASSIGNMENTS FOR PIN 1 ORIENTATION IN TAPE**


\*All dimensions are nominal

| Device        | Package Type | Package Drawing | Pins | SPQ  | Reel Diameter (mm) | Reel Width W1 (mm) | A0 (mm) | B0 (mm) | K0 (mm) | P1 (mm) | W (mm) | Pin1 Quadrant |
|---------------|--------------|-----------------|------|------|--------------------|--------------------|---------|---------|---------|---------|--------|---------------|
| TCA8418RTWR   | WQFN         | RTW             | 24   | 3000 | 330.0              | 12.4               | 4.3     | 4.3     | 1.3     | 8.0     | 12.0   | Q2            |
| TCA8418RTWR   | WQFN         | RTW             | 24   | 3000 | 330.0              | 12.4               | 4.25    | 4.25    | 1.15    | 8.0     | 12.0   | Q2            |
| TCA8418RTWRG4 | WQFN         | RTW             | 24   | 3000 | 330.0              | 12.4               | 4.25    | 4.25    | 1.15    | 8.0     | 12.0   | Q2            |

**TAPE AND REEL BOX DIMENSIONS**


\*All dimensions are nominal

| Device        | Package Type | Package Drawing | Pins | SPQ  | Length (mm) | Width (mm) | Height (mm) |
|---------------|--------------|-----------------|------|------|-------------|------------|-------------|
| TCA8418RTWR   | WQFN         | RTW             | 24   | 3000 | 356.0       | 356.0      | 36.0        |
| TCA8418RTWR   | WQFN         | RTW             | 24   | 3000 | 367.0       | 367.0      | 35.0        |
| TCA8418RTWRG4 | WQFN         | RTW             | 24   | 3000 | 367.0       | 367.0      | 35.0        |

## GENERIC PACKAGE VIEW

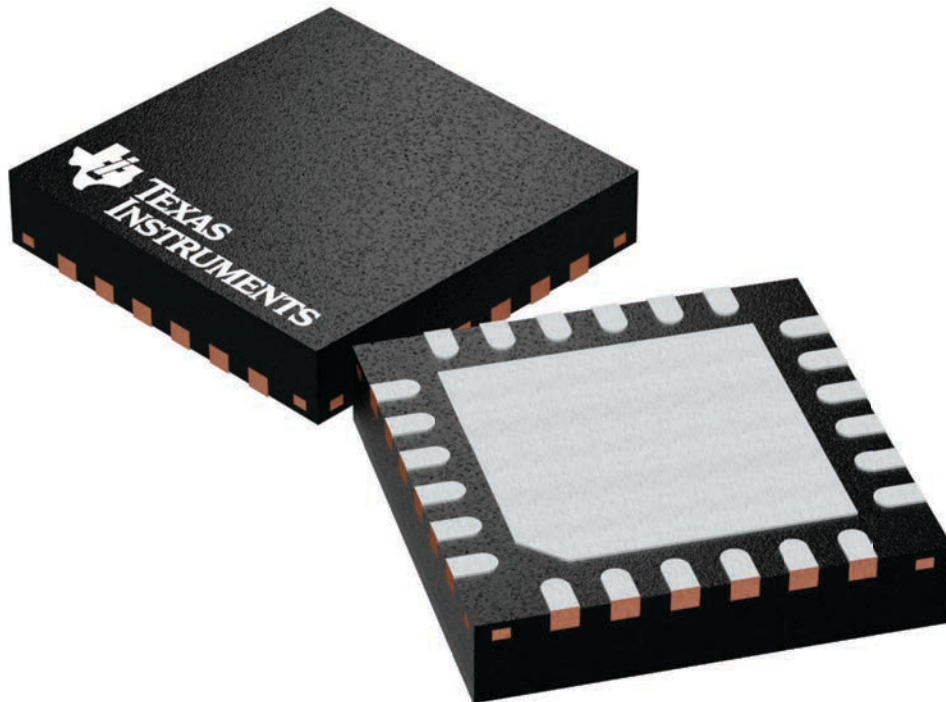
**RTW 24**

**WQFN - 0.8 mm max height**

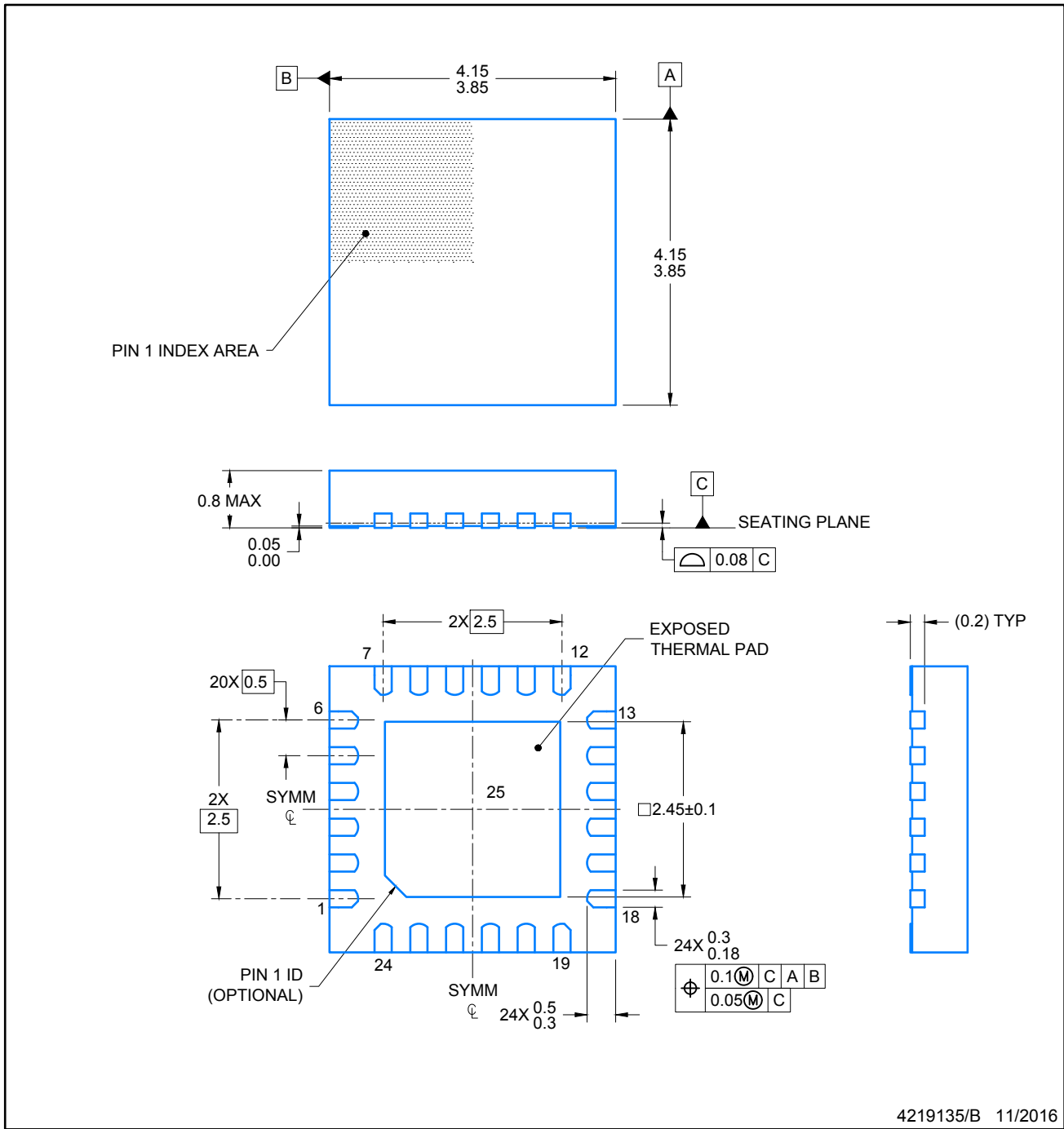
4 x 4, 0.5 mm pitch

PLASTIC QUAD FLATPACK - NO LEAD

This image is a representation of the package family, actual package may vary.  
Refer to the product data sheet for package details.

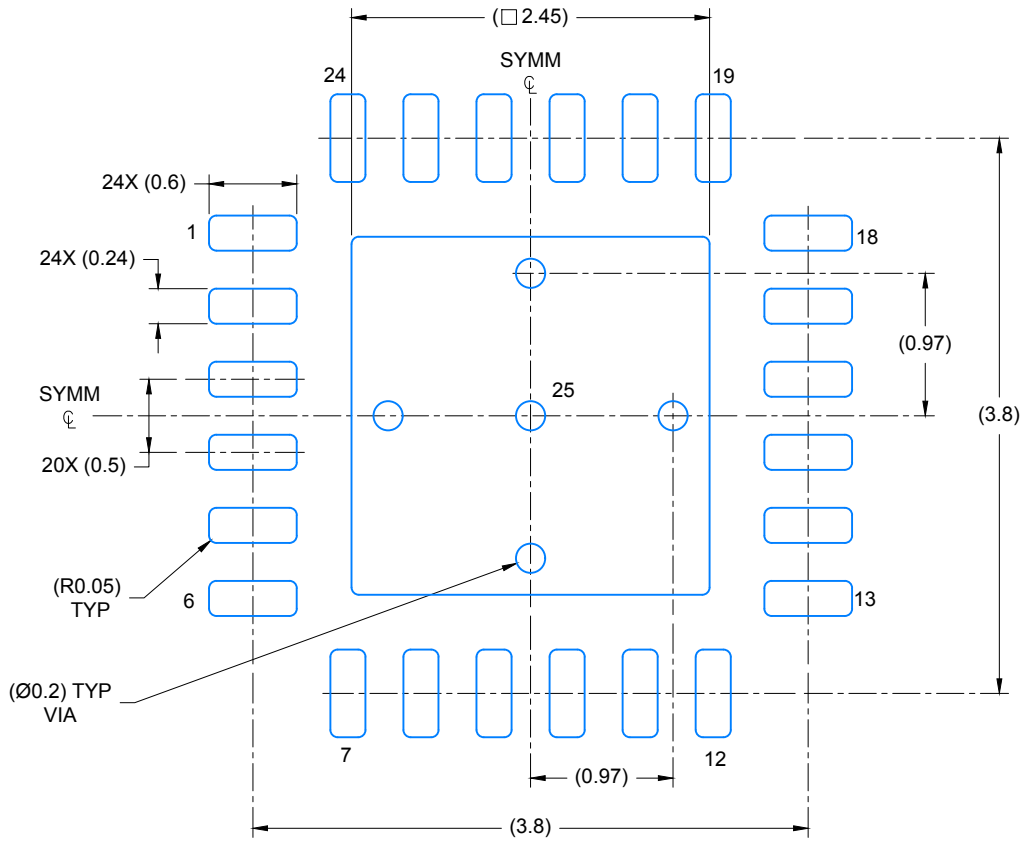


4224801/A

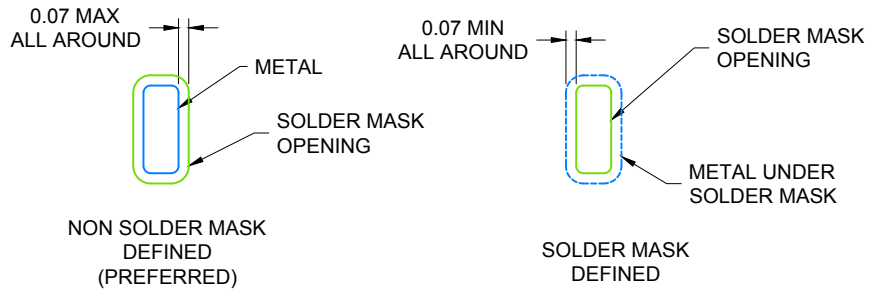


NOTES:

1. All linear dimensions are in millimeters. Any dimensions in parenthesis are for reference only. Dimensioning and tolerancing per ASME Y14.5M.
2. This drawing is subject to change without notice.



LAND PATTERN EXAMPLE  
SCALE: 20X

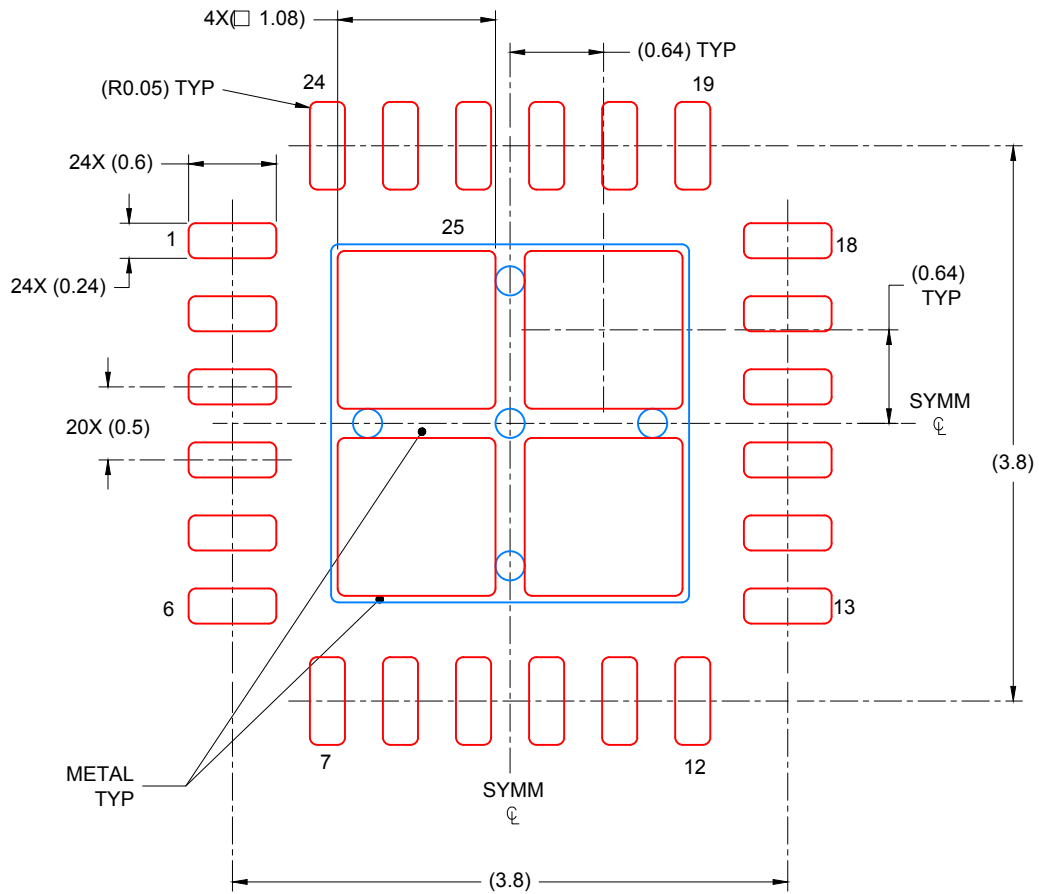


SOLDER MASK DETAILS

4219135/B 11/2016

NOTES: (continued)

- For more information, see Texas Instruments literature number SLUA271 ([www.ti.com/lit/sluea271](http://www.ti.com/lit/sluea271)).



SOLDER PASTE EXAMPLE  
 BASED ON 0.125 mm THICK STENCIL

EXPOSED PAD 25:  
 78% PRINTED COVERAGE BY AREA UNDER PACKAGE  
 SCALE: 20X

4219135/B 11/2016

NOTES: (continued)

4. Laser cutting apertures with trapezoidal walls and rounded corners may offer better paste release. IPC-7525 may have alternate design recommendations.

## 重要通知和免责声明

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

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

这些资源如有变更，恕不另行通知。TI 授权您仅可将这些资源用于研发本资源所述的 TI 产品的相关应用。严禁以其他方式对这些资源进行复制或展示。您无权使用任何其他 TI 知识产权或任何第三方知识产权。对于因您对这些资源的使用而对 TI 及其代表造成的任何索赔、损害、成本、损失和债务，您将全额赔偿，TI 对此概不负责。

TI 提供的产品受 [TI 销售条款](#)、[TI 通用质量指南](#) 或 [ti.com](#) 上其他适用条款或 TI 产品随附的其他适用条款的约束。TI 提供这些资源并不会扩展或以其他方式更改 TI 针对 TI 产品发布的适用的担保或担保免责声明。除非德州仪器 (TI) 明确将某产品指定为定制产品或客户特定产品，否则其产品均为按确定价格收入目录的标准通用器件。

TI 反对并拒绝您可能提出的任何其他或不同的条款。

版权所有 © 2026，德州仪器 (TI) 公司

最后更新日期：2025 年 10 月