TEXAS INSTRUMENTS

Application Report

SCPA032 - August 2001

Improving System Interrupt Management Using the PCF8574 and PCF8574A I/O Expanders for the I²C Bus

Emmanuel Tomdio Nana *Standard Linear & Logic*

ABSTRACT

In today's complex, embedded, computing and data-communication systems, interrupts are used extensively to service peripheral devices. However, because of the pin-number limitation on packages, most microprocessors have only one or two interrupt lines. Consequently, several devices usually are connected to the same interrupt line. The drawback to this configuration is that the overhead processing time to identify the device that requested the interrupt service may be too long (in the order of microseconds). The Texas Instruments PCF8574 and PCF8574A remote, 8-bit I/O expander for the I²C bus is designed for the specific application of monitoring a multiparallel bus. The remote I/O expander can inform the microprocessor if there is incoming data on the port or if there is a change of port data, without having to communicate via the I²C bus. This application report discusses the logic functionality of the PCF8574 and the PCF8574A, the specifics of the I²C interface, the electrical characteristics of the device, and its use in the intended applications. The information in this application report, along with the data sheet, should enable a system designer to successfully implement a parallel-bus monitor solution.

Contents

List of Figures

List of Tables

Introduction

The current trend in systems design is to reduce the number of copper traces on printed circuit boards (PCBs) required to route the address, data, and control lines, and to eliminate the many address decoders and glue-logic interface circuitry that connect to many devices. The reduction in copper traces renders the system less susceptible to disturbances by electromagnetic interference (EMI) and electrostatic discharge (ESD) and reduces system-debugging time. In addition, the elimination of the many address decoders and glue-logic interface circuitry saves board area, increases system reliability, and cuts cost.

The TI PCF8574 and PCF8574A accomplish both of these goals. This application report addresses the background, functionality, and application of these devices.

In this report, PCF8574 refers to the PCF8574 and PCF8574A devices, unless otherwise stated.

Background

Servers, workstations, backplanes, and personal computer (PC) systems that are designed to support Intel™, Motorola™, and Sun™ microprocessors typically have glue-logic interface circuitry and a basic monitor program that continually examines input lines from peripherals, such as a keyboard, temperature sensors, or other sensors. Such systems also allow the external user to communicate with the system. The different monitored lines are sometimes connected directly to the pins on the microprocessor, and the monitoring is done in software. This technique may require many pins on the microprocessor. Alternatively, to conserve pins on the microprocessor, the lines to be monitored communicate with the microprocessor through a glue-logic interface. In this case, fewer lines are connected to the microprocessor pins. In either case, development of such systems is more cumbersome and time consuming because more devices, more traces to be routed, or more complex code development is required.

The PCF8574 has a two-wire communication bus that the microprocessor monitors for each PCF8574 device (up to eight lines) that is connected to the PCF8574 parallel input/output (I/O) port (P port).

The TI PCF8574 also has an interrupt line ($\overline{\text{INT}}$) that can be connected to the interrupt logic of the microprocessor. By sending an interrupt signal on this line, the remote I/O informs the microprocessor that there is incoming data or a change of data on its ports without having to communicate via the communication bus.

Furthermore, the TI PCF8574 has low current consumption and includes latched outputs with high current drive capability for directly driving LEDs. The devices are designed to provide general-purpose, remote I/O expansion for most microprocessor families through the two-wire bidirectional interintegrated circuit (I^2C) bus. The I^2C bus consists of two active wires and a ground connection. The active wires are the bidirectional serial data (SDA) line and the bidirectional serial clock (SCL) line. However, for the PCF8574, the SCL line is unidirectional, from the microprocessor to the device.

All this functionality is fully integrated into the TI PCF8574, a remote, 8-bit I/O expander. By using the PCF8574, circuit designers can greatly simplify complex glue-logic interface circuitry, reduce PCB area used, and improve system reliability.

I^2C Protocol Overview

Every device connected to the I^2C bus has its own unique address and can act as a receiver and/or transmitter, depending on its functionality. The I^2C bus is a multimaster bus, meaning that there could be one or more bus masters. The bus master issues the command on the bus that initiates data transfer, and the bus master provides the SCL signal for data transfer over the I^2C bus. At that time, all the other devices on the bus are regarded as bus slaves.

Typically, a complete I^2C communication consists of a start (or repeated start) condition; a slave-address transfer, followed by a data-direction bit; a receiver-acknowledge bit; one or more data-byte transfers, each followed by a receiver-acknowledge bit; and a stop condition (see Figure 1).

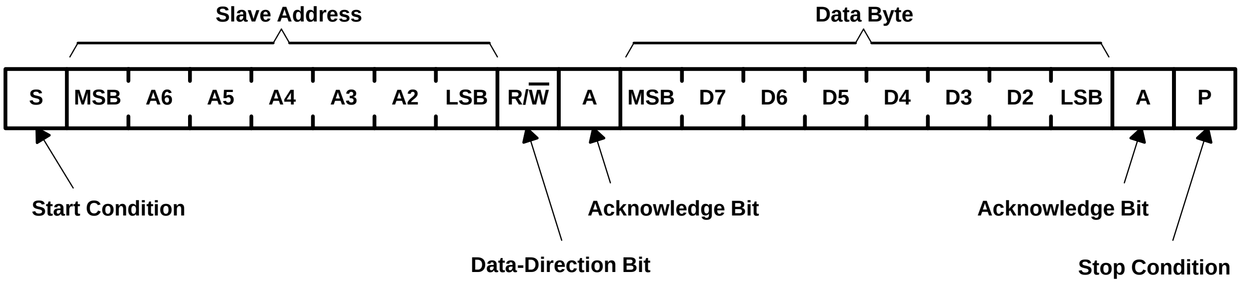

Figure 1. Complete I^2C Communications

A start (or repeated start) condition is a high-to-low transition on the SDA I/O while the SCL input is high (see Figure 2). A stop condition is a low-to-high transition on the SDA I/O while the SCL input is high (see Figure 2). All other valid SDA transitions must occur only while SCL is low.

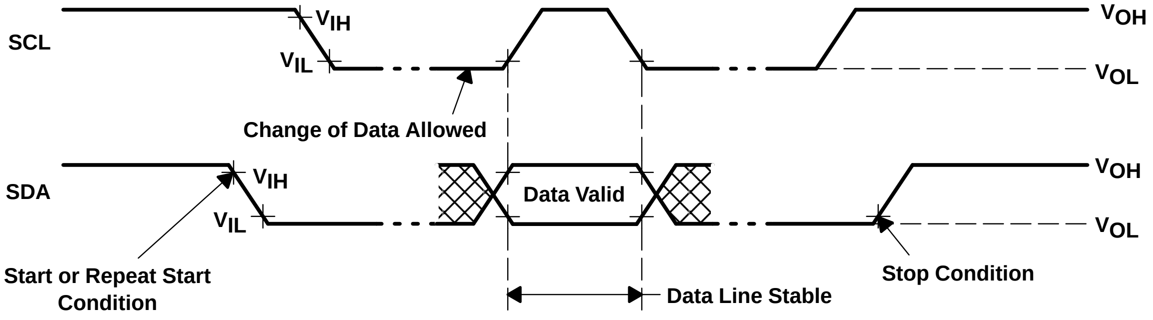

Figure 2. I^2C Signals

Address and data transfers are composed of 8-bit bytes. Each byte transfer is followed by an acknowledge (ACK) clock cycle. The device that receives the data transfer should acknowledge the receipt. In some cases this can take the form of no acknowledge (NACK). The address transfer usually is a 7-bit word (the I^2C bus protocol also supports 10-bit addressing) followed by the read/$\overline{\text{write}}$ (R/$\overline{W}$) data-direction bit.

If a device controls the SCL line, it is considered to be a master. It initiates a data transfer by sending a start condition followed by an address word and $R/\overline{W}$ bit. The device that acknowledges the address sent by the master is considered the slave. The direction of data transfer on the bus determines whether the master is a receiver or transmitter. The same is true for the slave. If the $R/\overline{W}$ bit is low, a master has indicated a write transfer and is considered a master transmitter. The slave, whose address was sent by the master, acknowledges and becomes a slave receiver. Similarly, if the $R/\overline{W}$ bit is high, the master requested a read transfer and is considered a master receiver. The slave, whose address was sent by the master, acknowledges and becomes a slave transmitter.

For additional information on the I^{2}C protocol, refer to *The I^{2}C Bus and How to Use It*.[1]

Device Description

Logic Functionality

Communication with the PCF8574 is initiated by the master sending a start condition. A high-to-low transition on the SDA I/O while the SCL input is high (1) initiates I^{2}C communication with the device. After the start condition, the device address byte is sent, most significant bit (MSB) first, including the data-direction bit ($R/\overline{W}$). Table 1 shows a format of the address byte.

Table 1. PCF8574 and PCF8574A Address Byte

DEVICE	BIT							
	7 (MSB)	6	5	4	3	2	1	0 (LSB)
PCF8574 address	0	1	0	0	A2	A1	A0	$R/\overline{W}$
PCF8574A address	0	1	1	1	A2	A1	A0	$R/\overline{W}$

An address bit, a data bit, or an ACK bit is transferred on the SDA line during an SCL cycle. Data on the SDA line must remain stable during the high period of the clock pulse because changes in the data line, at this time, are interpreted as control signals (start or stop).

After receiving the valid address byte, the device responds with an ACK, which is a low (0) on the SDA I/O during the high on the ACK-related clock pulse.

The data byte transfer follows the address ACK. If the $R/\overline{W}$ bit is high, the device has control of the SDA line and the data from the device is the data read from the P port. Each data byte transfer is followed by an ACK sent to the device. After the ACK, if the device receives a no-stop condition, it recaptures the P port at the rising edge of the ACK cycle. Then, if the device receives additional clock pulses, it outputs the second data byte captured from the P port. However, if the device receives a NACK during the ACK cycle and receives a no-stop condition, it does not recapture the P port.

If the $R/\overline{W}$ bit is low, the data bits on the SDA line are sent to the device to be output to the P port. The data byte is followed by an ACK sent from the device. Data are output only if complete bytes are received and acknowledged. The output data is valid after the low-to-high transition of SCL, during the clock cycle for the ACK. If other data bytes are sent to the device following the ACK, the device sends an ACK and updates the P port again.

The master sends a stop condition, which is a low-to-high transition on the SDA I/O while the SCL input is high.

Each of the PCF8574 eight I/Os can be used independently as an input or output. Input data is transferred from the port to the microprocessor by the read mode, while output data is transferred from the microprocessor to the port by the write mode. Before using the I/O as an input, a high must be written to the I/O first.

I^{2}C Interface

The PCF8574 cannot control the SCL line. The PCF8574 SCL line is for input only and the PCF8574 is, therefore, always considered a slave device. Because the PCF8574 is capable of both receiving and transmitting data, it can be a slave transmitter or a slave receiver, depending on the state of the R/$\overline{W}$ bit in the address transfer.

I^{2}C communication with the PCF8574 is initiated by a master sending a start condition (see Figure 3). Following a start condition, a device address byte is received from the master device, MSB first, including the R/$\overline{W}$ bit. The first part of the address byte consists of a 4-bit address code, which is set to 0100 for the PCF8574 device and 0111 for the PCF8574A device. Three chip-select bits (A2, A1, and A0) follow the address code. The chip-select bits allow the use of up to eight PCF8574 devices and eight PCF8574A devices on the same bus, and are used to determine which device is accessed. The chip-select bits in the address byte must correspond to the logic levels on the corresponding A2, A1, and A0 pins for the device to respond. After receiving the valid address byte, the PCF8574 responds with an ACK, which is a low on the SDA I/O during the high of the ACK-related clock pulse.

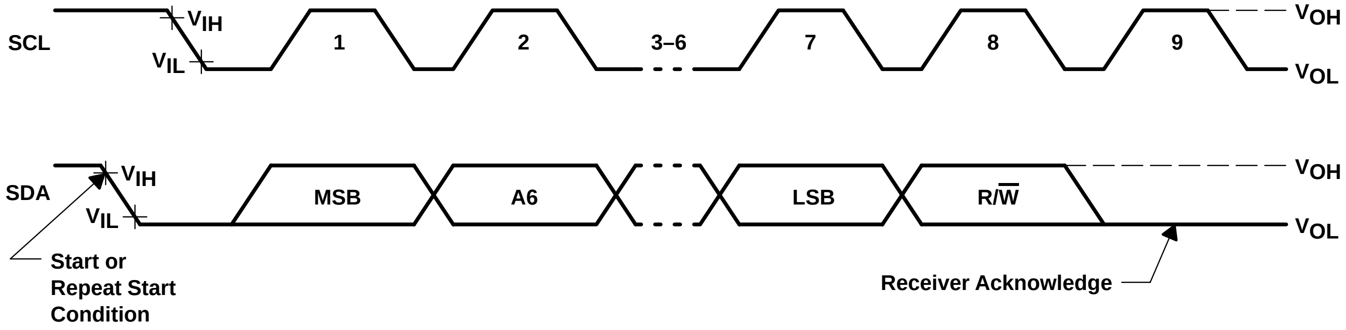

Figure 3. I^{2}C Address Transfer

The data byte follows the address ACK (see Figure 4). If the R/$\overline{W}$ bit is high, data from the PCF8574 are the values read from the 8-bit I/O port. If the R/$\overline{W}$ bit is low, the data from the master are written to the 8-bit I/O port.

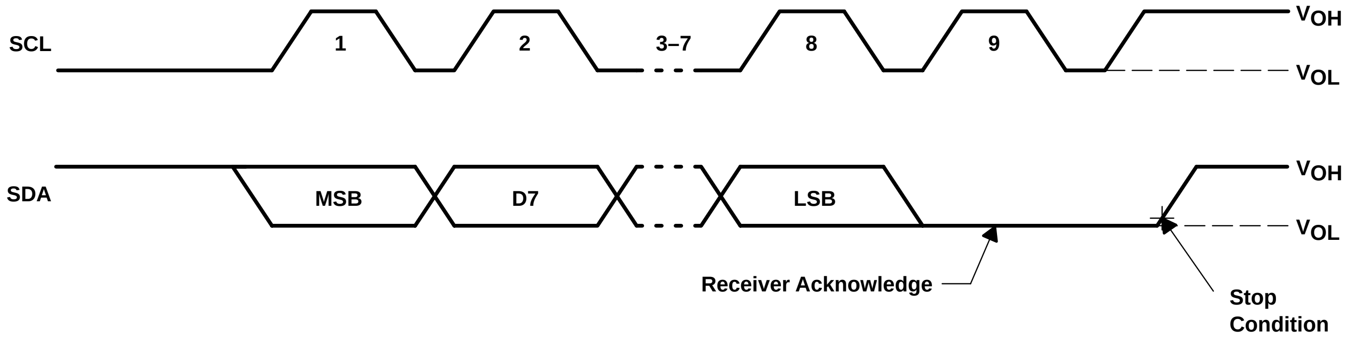

Figure 4. I^{2}C Data Transfer

The master sends a stop condition (see Figure 4) to stop the transfer of data by transitioning the SDA line from low-to-high while SCL is high. When this occurs, the data present at the last acknowledge phase is valid (output mode); input data is lost.

Addressing

Each device on the I^{2}C bus must be configured to have a unique address. The first four bits of the 7-bit address for the PCF8574 are 0100 and those for the PCF8574A are 0111. The lower three bits are the settings on A2, A1, and A0. The complete unique address of a device is determined by the setting on the A2, A1, and A0 pins. Table 2 shows the unique addresses of the PCF8574 and PCF8574A devices for the different possible settings.

Table 2. PCF8574 and PCF8574A A[2:0] I^{2}C Bus Slave Address Map

INPUTS			PCF8574 I^{2}C BUS SLAVE ADDRESS	PCF8574A I^{2}C BUS SLAVE ADDRESS
A2	A1	A0		
L	L	L	32 (decimal), 20 (hexadecimal)	56 (decimal), 38 (hexadecimal)
L	L	H	33 (decimal), 21 (hexadecimal)	57 (decimal), 39 (hexadecimal)
L	H	L	34 (decimal), 22 (hexadecimal)	58 (decimal), 3A (hexadecimal)
L	H	H	35 (decimal), 23 (hexadecimal)	59 (decimal), 3B (hexadecimal)
H	L	L	36 (decimal), 24 (hexadecimal)	60 (decimal), 3C (hexadecimal)
H	L	H	37 (decimal), 25 (hexadecimal)	61 (decimal), 3D (hexadecimal)
H	H	L	38 (decimal), 26 (hexadecimal)	62 (decimal), 3E (hexadecimal)
H	H	H	39 (decimal), 27 (hexadecimal)	63 (decimal), 3F (hexadecimal)

The ability to set unique addresses for the devices makes it possible to have up to a maximum of eight PCF8574 and eight PCF8574A devices on the same I^{2}C bus.

The address written to the device is actually compared with the programmed address at the falling edge of the $R/\overline{W}$ clock cycle. If the address is slightly changed just before the falling edge of the $R/\overline{W}$ clock cycle, there is no ACK. If the master sends a start condition and the correct address of the device (as programmed initially), the programmed address changes before the falling edge of the $R/\overline{W}$ bit clock cycle. Although the correct address had been received serially, the slave does not respond with any ACK.

Similarly, if the master sends a start condition and another address of the device (not matching the initially programmed address), the programmed address changes to match the address received serially before the falling edge of the $R/\overline{W}$ bit clock cycle, and the slave responds with an ACK.

Interrupt

The PCF8574 provides an open-drain interrupt ($\overline{INT}$) output that can be fed to a corresponding input of the microprocessor. This gives the PCF8574 the capability to initiate an action in a system.

Logically, $\overline{INT}$ is the output of a comparator that exclusively ORs the I/O port pins (D inputs into the P-read D flip-flops) and the corresponding Q outputs on the P-read D flip-flops. Figure 5 shows data flow from the SDA I/O to the P-port I/O, including the interrupt logic.

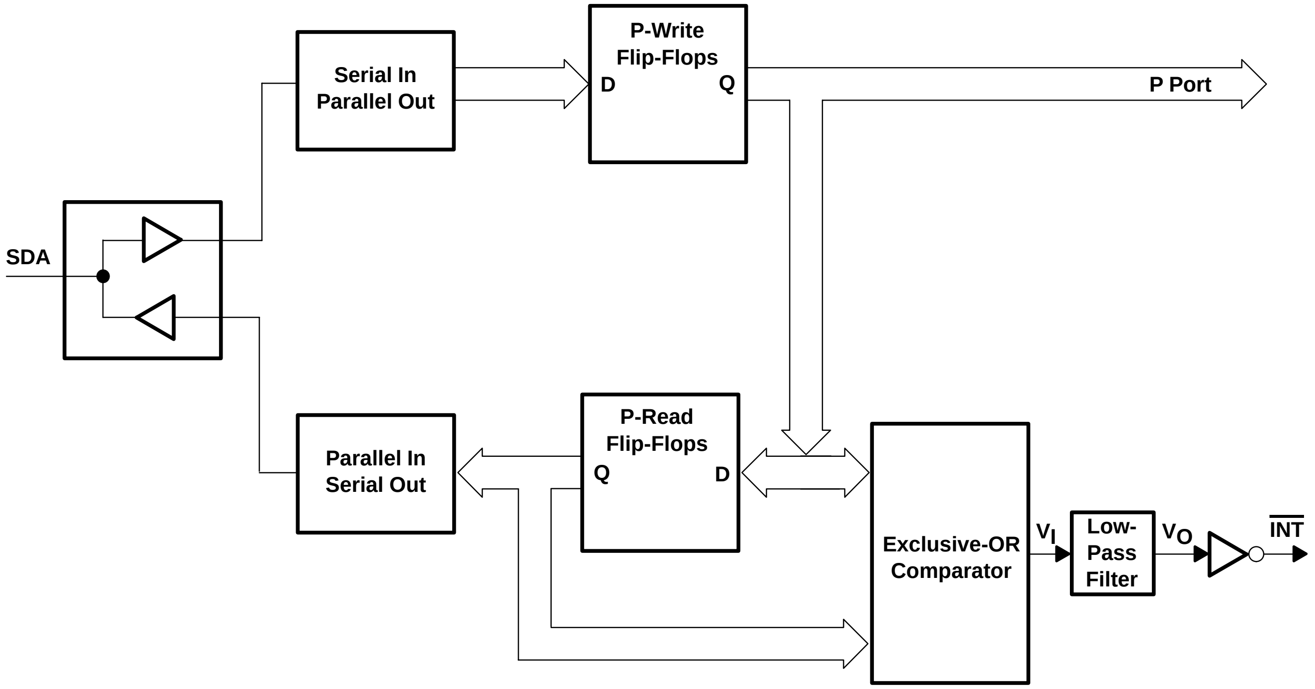

Figure 5. Data Flow and Interrupt Logic

When data at the I/O port changes before data on the I/O port is read in at the P-read port, data at the D input of the P-read port is different from data at the Q output of the P-read port. In this case, exclusive ORing the data at the D input of the P-read port with data at the Q output of the P-read port asserts $\overline{INT}$.

During the write operation, $\overline{INT}$ is not affected by a change in data at the P port. In the write mode after the master sends a data byte to the slave, the slave acknowledges on the falling edge of the eighth data bit clock cycle. The slave then updates the P port on the rising edge of the ACK clock cycle and updates the P-read port on the falling edge of the ACK clock cycle. There is, therefore, a time interval between which the data at the P port (D inputs into the P-read D flip-flops) and the data at the corresponding Q outputs on the P-read D flip-flops are different. However, during the entire write cycle, $\overline{INT}$ is not affected as a result of this change in data at the P port. This is so because $\overline{INT}$ is disabled (high) externally for the data ACK cycle interval that the difference in data at the input of the P-read port and data at the output of the P-read port exits.

During a read operation, $\overline{\text{INT}}$ can be asserted in the data ACK cycle if the data read is different from the data at the PCF8574 output. Unlike the write operation, during the read operation, $\overline{\text{INT}}$ is disabled (high) externally only in the address ACK cycle and not in the data ACK cycle. Therefore, any difference in the data at the P port (D inputs into the P-read D flip-flops) and the data at the corresponding Q outputs on the P-read D flip-flops during the data ACK cycle results in $\overline{\text{INT}}$ being asserted.

Resetting and reactivating the interrupt circuit is achieved when data on the port is changed to the original setting or data is read from, or written to, the port that generated the interrupt. In the case where the data on the port is changed to the original setting, data at the D input of P-read flip-flops equals data at the Q output of the P-read flip-flops. Hence, exclusive ORing the two signals causes $\overline{\text{INT}}$ to be deasserted. Likewise, reading the data from, or writing to, the port leads to the data at the D input of P-read flip-flops to equal the data at the Q output of P-read flip-flops. Hence, $\overline{\text{INT}}$ is deasserted.

Specifically, in the read mode, deasserting $\overline{\text{INT}}$ occurs at the ACK bit, after the low-to-high transition of SCL. This is so because, in the read mode, it is on the rising edge of SCL during the ACK cycle that the P-port input register is loaded with the input data. Furthermore, in the write mode, deasserting $\overline{\text{INT}}$ occurs at the ACK bit, after the high-to-low transition of SCL. This is so because, in the write mode, it is on the falling edge of SCL during the ACK cycle that the P-port output register is updated with the output data.

Each change of the I/Os after resetting is detected and, after the next rising clock edge, is transmitted as $\overline{\text{INT}}$. Reading from, or writing to, a device does not affect the interrupt circuit of other devices on the same I^{2}C bus.

If a change occurs at the I/O port and one of the corrective measures is taken within 420 ns, the $\overline{\text{INT}}$ line is not asserted, not even for the short period that a difference exists between the data on the D input and Q output of the P-read port flip-flops. This is due to the low-pass filter at the output of the comparator.

Internal Operation

The PCF8574 consists of an 8-bit I/O port connected to the I^2C bus through a shift register. The simplified schematic diagram of each P-port I/O from the shift register to the I/O port is shown in Figure 6.

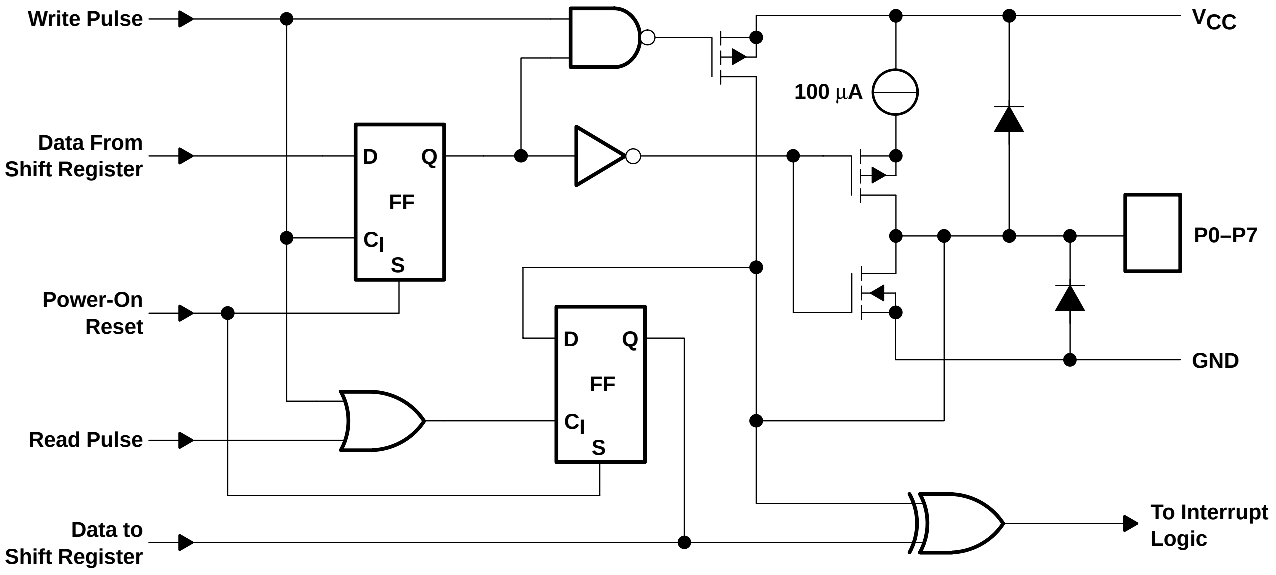

Figure 6. Simplified Schematic Diagram of Each P-Port I/O

In the write mode, the parallel output port register is loaded on the rising edge of SCL during the ACK cycle, and the parallel input port register is loaded on the falling edge of SCL during the ACK cycle. When the parallel port input register is loaded, the parallel input/compare register is captured from the P port during the falling edge of SCL during the ACK cycle, and the interrupt flag is updated internally. However, $\overline{\text{INT}}$ is not active for a full SCL cycle. Figure 7 shows the write operation.

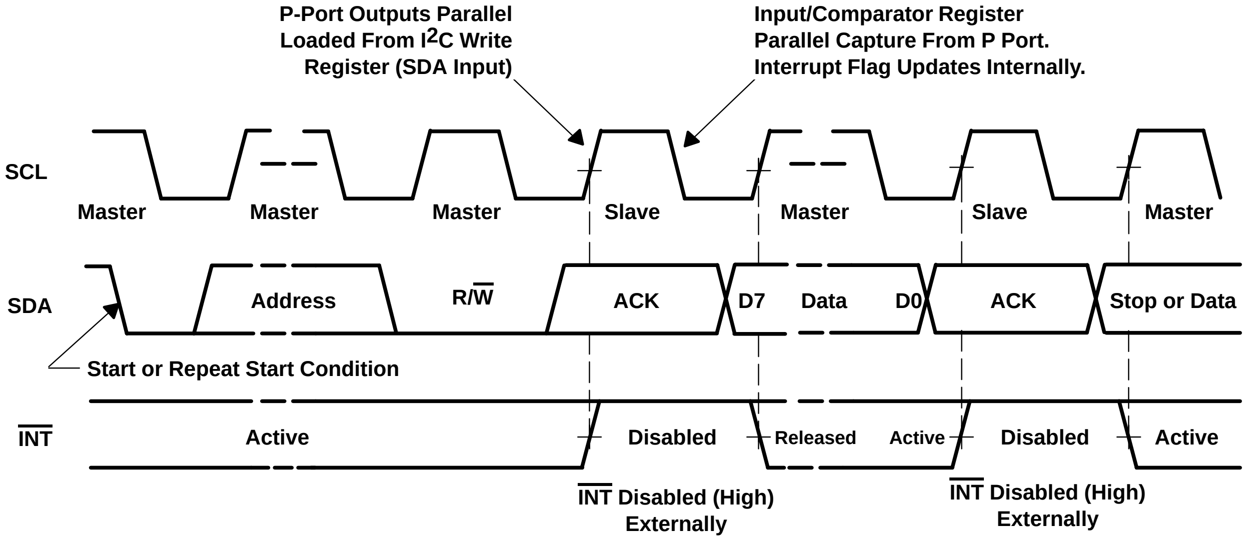
P-Port Outputs Parallel Loaded From I2C Write Register (SDA Input)
Input/Comparator Register Parallel Capture From P Port. Interrupt Flag Updates Internally.
SCL
Master
Master
Master
Slave
Master
Slave
Master
SDA
Address
R/W
ACK
D7
Data
D0
ACK
Stop or Data
Start or Repeat Start Condition
INT
Active
Disabled
Released
Active
Disabled
Active
INT Disabled (High) Externally
INT Disabled (High) Externally

Figure 7. Write Operation

There are two ACK cycles for the write cycle, and both ACK responses are provided by the PCF8574 (slave). Note that $\overline{INT}$ is disabled during both ACK cycles.

In the write mode, the PCF8574 can support multiple data bytes. The protocol begins by the master sending a start bit and an address byte with R/$\overline{W}$ low. After the slave acknowledges, the master sends the data byte and waits for an ACK. After each ACK from the slave, the master sends another data byte. During multiple data-byte writes, the PCF8574 updates the P port to the data received from the master, on the rising edge of the related ACK clock cycle.

In the read mode, the parallel input port register is loaded on the rising edge of SCL during the ACK cycle. The P-read shift register is parallel loaded on the falling edge of SCL during the ACK cycle (MSB falls to output). When the parallel port input register is loaded, the parallel input/compare register is captured from the P port during the rising edge of SCL during the ACK cycle, and the interrupt flag is updated internally. However, $\overline{INT}$ is not active for a full SCL cycle. Figure 8 shows the read operation.

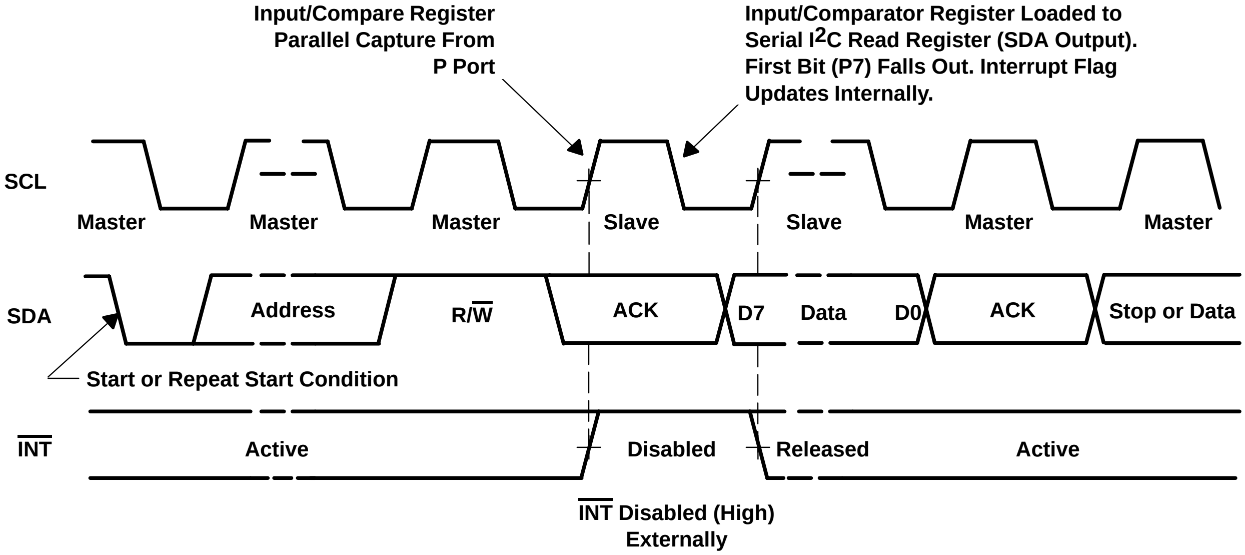
Input/Compare Register Parallel Capture From P Port
Input/Comparator Register Loaded to Serial I2C Read Register (SDA Output). First Bit (P7) Falls Out. Interrupt Flag Updates Internally.
SCL
Master
Master
Master
Slave
Slave
Master
Master
SDA
Address
R/W
ACK
D7
Data
D0
ACK
Stop or Data
Start or Repeat Start Condition
INT
Active
Disabled
Released
Active
INT Disabled (High) Externally

Figure 8. Read Operation

There are two ACK cycles for the read cycle. The PCF8574 (slave) provides the first (address) ACK and the master provides the second (data) ACK. The data ACK, which is from the master, is optional. Note that $\overline{INT}$ is disabled only during the address ACK cycle.

The PCF8574 supports multiple-byte reads. During the read cycle, the master sends a start condition and address (with R/$\overline{W}$ bit set to read). After the slave acknowledges, the slave transmits the captured data byte. For multiple-byte read, the master must then acknowledge the data byte. If the master for a data byte sends NACK, the PCF8574 does not capture the P port again and the device does not transmit data again.

During the read and write operations, $\overline{INT}$ is disabled when the PCF8574 (slave) provides the ACK. Interrupts, which occur during these ACK clock pulses, can be lost (or be very short), due to the resetting of $\overline{INT}$ during the ACK cycle.

For proper operation, before using an I/O pin as an input, a high must be written to the I/O. The P port has weak pullups that allow for fast rising edges into heavily loaded outputs. These devices turn on when an output is written high, and are switched off by the negative edge of SCL. Therefore, if the PCF8574 outputs are all high (either from power-up reset initialization or from a write) before a read is performed, any external driver connected to the I/O can fully control the I/O. If the PCF8574 output is written low before a read, a low is always read from the I/O, regardless of the state of the external driver that is connected to the I/O.

Electrical Characteristics

AC Performance

The TI PCF8574 is designed to meet or exceed the I^2C bus fast-mode ac performance requirements. System management bus (SMBus) is a similar two-wire bus protocol based on the I^2C bus. In most cases, fast-mode I^2C ac specifications exceed the requirements of the SMBus specification, which can be met with most of the I^2C bus standards mode requirements. Table 3 provides a comparison of the ac specification of the I^2C bus fast-mode limits versus the SMBus limits, along with the PCF8574 capability.

Table 3. Comparison of AC Specifications Between I^2C Protocol and SMBus Revision 1.1

SYMBOL	PARAMETER	I^2C BUS (FAST MODE)		SMBus		UNIT	PCF8574[†]
		MIN	MAX	MIN	MAX		
f_{BUS}	Operating frequency	0	400	10	100	kHz	Yes
t_{BUF}	Bus free time between stop and start condition	1.3		4.7		µs	Yes
t_h; STA	Hold time after (repeated) start condition	0.6		4		µs	Yes
t_{su}; STA	Repeated-start condition setup time	0.6		4.7		µs	Yes
t_{su}; STO	Stop condition setup time	0.6		4		µs	Yes
t_h; DAT	Data hold time	0[‡]		300		ns	Yes
t_{su}; DAT	Data setup time	100		250		ns	Yes
$t_{TIMEOUT}$	Clock low timeout value	N/A		25	35	ms	No[§]
t_{LOW}	Clock low period	1.3		4.7		µs	Yes
t_{HIGH}	Clock high period	0.6		4	50	µs	Yes
t_{LOW}; SEXT	Cumulative clock, low extend time (slave device)	N/A			25	ms	Yes[¶]
t_{HIGH}; MEXT	Cumulative clock, low extend time (master device)	N/A			10	ms	N/A
t_f	Clock/data fall time		300		300	ns	Yes
t_r	Clock/data rise time		300		1000	ns	Yes

[†] The comparison in this column is to determine if the TI PCF8574 meets or exceeds the requirements of the SMBus specification Revision 1.1.

[‡] The I^2C bus protocol requires that a device must internally provide a hold time of at least 300 ns.

[§] The PCF8574 does not support timeout.

[¶] The PCF8574 does not require the extension of the bus clock for operation. The PCF8574 can send and receive at a rate up to 400 kHz with t_{LOW} = 1.3 µs minimum and t_{HIGH} = 0.6 µs minimum. However, if a proper write-mode sequence is issued from a master device, the PCF8574 begins writing to the I/O port after receiving the stop condition. The specified typical write time (t_{WT}) is 10 ms. If the device is addressed during this required programming time, the device does not acknowledge. By operating in this fashion, the device does not tie up the bus during a program.

DC Performance

The TI PCF8574 is designed to meet or exceed the I^2C bus dc specifications. The PCF8574 SCL input and SDA I/O implement CMOS circuitry. Table 4 provides a comparison of the dc specifications of the I^2C bus (fast mode) limits for V_{CC}-related inputs versus the SMBus Revision 1.1 limits, along with the PCF8574 capability.

Table 4. Comparison of DC Specifications Between I^2C Protocol and SMBus Revision 1.1, V_{CC}-Related Inputs

SYMBOL	PARAMETER	I^2C BUS (FAST MODE)		SMBus		UNIT	PCF8574[†]
		MIN	MAX	MIN	MAX		
V_{IL}	Data, clock input – low-level voltage	–0.5	$0.3 \times V_{CC}$	–0.5	0.8	V	Yes
V_{IH}	Data, clock input – high-level voltage	$0.3 \times V_{CC}$	$V_{CC} + 0.5$	2.1	5.5	V	Yes
V_{OL}	Data, clock output – low-level voltage	0	0.4[‡]		0.4[§]	V	Yes[¶]
I_{lkg}	Input leakage current	N/A	N/A		±5	µA	No
I_{PULLUP}	Current through pullup resistor or current source	N/A	N/A	100	300	µA	N/A[#]

[†] The comparison in this column is to determine if the TI PCF8574 meets or exceeds the requirements of the SMBus specification Revision 1.1.

[‡] Maximum V_{OL} specified for 3-mA sink current

[§] Maximum V_{OL} specified for I_{PULLUP}MAX

[¶] The PCF8574 is a slave device, therefore, it does not control the SCL port. V_{OL} is applicable only to the SDA port.

[#] This specification is applicable to removable SMBus devices, such as a smart battery.

Application Circuitry

In the interface between a microprocessor and the microprocessor’s peripherals, the PCF8574 replaces several glue-logic interface circuitry and passive components in the system. Together with the TI THMC10 as a peripheral, the PCF8574 can be used for remote temperature monitoring and management in PCs, servers, workstations, backplanes, etc. Figure 9 shows an example remote temperature-monitoring and control application.

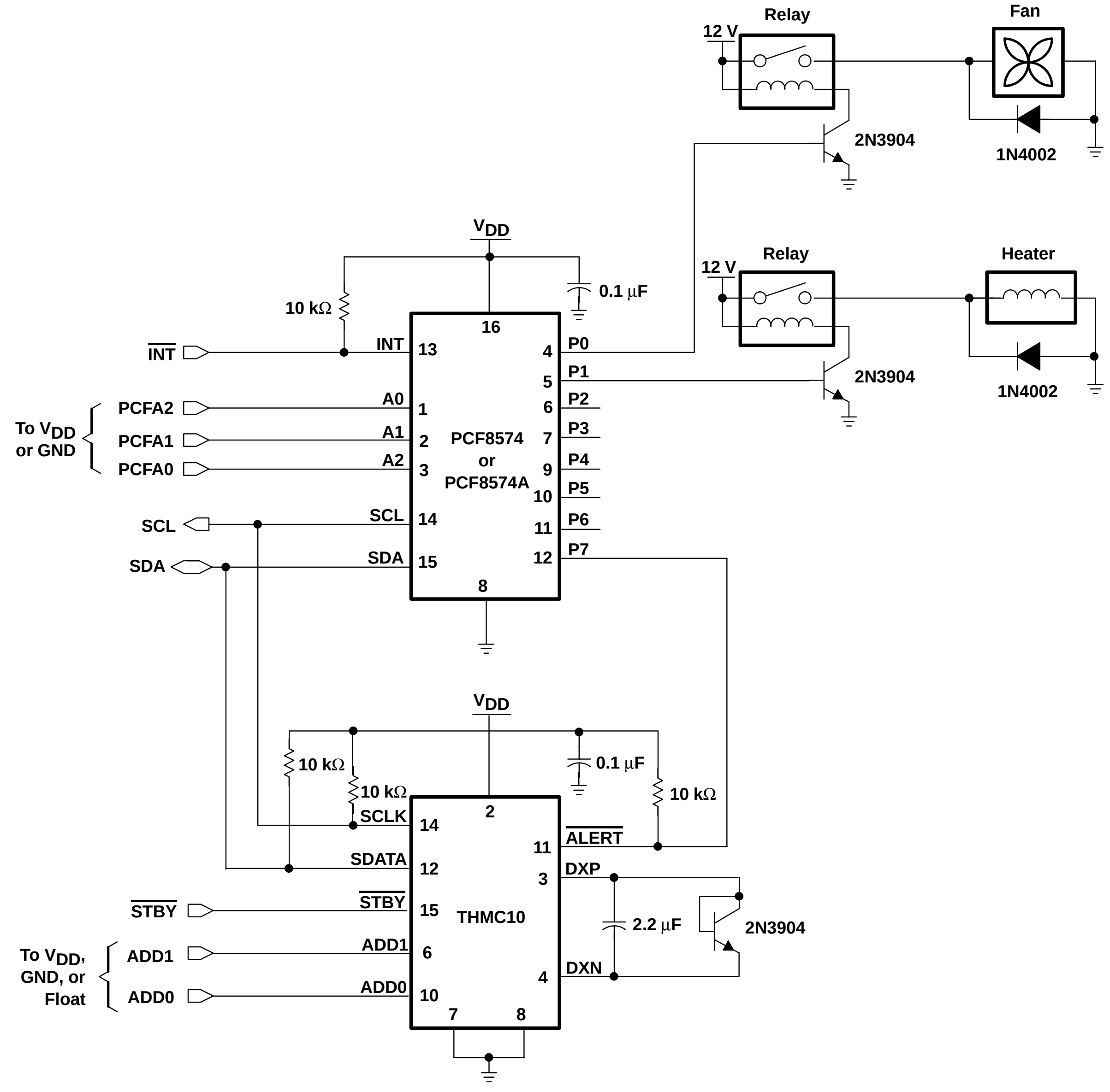

Figure 9. A PCF8574 Example Application Circuit for Monitoring and Managing System Temperature

In the remote temperature-monitoring and management application, the I^{2}C SCL and I^{2}C SDA pins of the PCF8574 are connected to open-drain I/Os on the microprocessor. External pullup termination resistors are required on the I^{2}C bus. The value of the termination depends on the capacitive loading of the bus.

The THMC10 is a dual, digital, temperature monitor with active-low under/over temperature alerts ($\overline{\text{ALERT}}$). It uses a two-wire SMBus interface to report temperature in an 8-bit, 2s-complement format in degrees Celsius (°C). The device is designed to measure the temperature of a microprocessor using a diode-connected transistor on the microprocessor die, such as the one present on the Intel Pentium II and III, and Sun UltraSPARC™ processors. In this remote temperature-sensing application, the THMC10 is used with a low-cost, diode-connected, discrete transistor, such as the 2N3904 or 2N3906.

Outputs of the parallel I/O pins from the peripherals are fed directly to the I/O pins of the PCF8574. The corresponding I^{2}C bus pins on the PCF8574 are connected with the two-wire SMBus interface pins on the THMC10. Under-temperature and over-temperature limits for the on-chip and remote temperature sensors are user programmable via the SMBus interface. The THMC10 is programmed to issue an alert when the monitored temperature falls below, or rises above, the user-programmed under-temperature or over-temperature limits. When either limit is exceeded, $\overline{\text{ALERT}}$ is asserted.

The $\overline{\text{ALERT}}$ pin is connected to an I/O pin (P7) of the PCF8574. When the data on the P port changes because $\overline{\text{ALERT}}$ is asserted, the PCF8574 interrupts the microprocessor through the $\overline{\text{INT}}$ line and the microprocessor reads the temperature of the THMC10 through the SMBus.

An I/O pin (P0) of the PCF8574 is also connected to the base of a transistor (Q1), which switches a relay on and off to drive a 12-V fan. Another I/O pin (P1) is connected to the base of another transistor (Q2), which switches a relay on and off to drive a 12-V heater. By writing the appropriate value to the PCF8574, the fan or the heater, or both, can be turned on or off, thereby regulating the temperature.

This application example takes into consideration only one fan and one heater peripheral. The SMBus can support a maximum of nine THMC10 devices and the I^{2}C bus can support a maximum of 16 PCF8574 or PCF8574A devices. Therefore, more than two peripherals can be connected, and the temperatures at different locations in a system can be monitored and controlled.

The information on the THMC10 presented in this application report is very basic to understanding the application circuit presented. For further information on the functional description, characteristics, and principles of operation of the THMC10, refer to the THMC10 data sheet (SLIS089).

Conclusion

The TI PCF8574 provides a simple, cost-effective method of monitoring, reading, or writing to a parallel bus. The I^{2}C interface is used to write data to the I/O port of the PCF8574 and read data on the I/O port of the PCF8574. The open-drain $\overline{\text{INT}}$ output, which can be connected to a corresponding input of the microprocessor, provides a mechanism for the PCF8574 to initiate action on the microprocessor without communicating through the I^{2}C bus. This gives the PCF8574 the capability to initiate an action in a system.

Frequently Asked Questions (FAQs)

Question 1: What are the PCF8574 and PCF8574A?

Answer: The PCF8574 and PCF8574A are designed to provide general-purpose remote I/O expansion for most microprocessor families through a two-wire bidirectional bus (I^2C).

Question 2: What is I^2C?

Answer: I^2C is an acronym for Inter-Integrated Circuit bus. Philips Semiconductors developed the I^2C bus in the early 1980s. Its purpose is to provide a communication link between a CPU and peripheral integrated circuit (IC). An I^2C bus physically consists of two active wires and a ground connection. The active wires are the serial data (SDA) line and serial clock (SCL) line, both of which are bidirectional.

Question 3: In a PCF8574 application, the slave does not respond to the master, and the data written to the slave is not output on the I/O port. Why is this so?

Answer: The slave does not respond to the master and output the data written to it if the I^2C protocol is violated. Either the proper I^2C start condition is not issued, or the I^2C communication is not terminated properly.

After every start condition, the full device address must be sent to the slave. If, in the application, the master sends a start condition and begins to send the slave address but does not send the complete slave address, the master sends another start condition and sends the remaining address bits, the device does not recognize the complete address. The device does not respond with an ACK because it did not get the full address after the second start condition. Similarly, the data written to the device is not output to the I/O port because the device did not get a proper write cycle.

Also, after every stop condition, the master must reissue the start condition, then send the full device address (with the $R/\overline{W}$ bit set to write) to the slave before sending any data. If, in the application, the master sends a start condition and begins to send the slave address, then sends the stop condition, the slave does not recognize any data that is sent after the stop condition. Consequently, the slave does not respond with an ACK because it did not get any start condition and the full address after the stop condition was issued. Similarly, data written to the device is not output to the I/O port because the device did not get a proper write cycle.

Question 4: Can the master write to some of the I/O pins on the PCF8574 by sending a start condition and the address byte (with the R/$\overline{W}$ bit set to write) and, after the device acknowledges, can the master send an incomplete byte to the PCF8574?

Answer: In order to write to the I/O pins on the PCF8574, the master must send a start (or repeated start) address byte (with the R/$\overline{W}$ bit set to write), and after the device acknowledges, the master must send a complete 8-bit byte to the slave. The slave does not acknowledge the data and it does not update the I/O port until it receives eight bits of data from the master.

Question 5: When the master writes to the device, the device outputs the data to the P port and, a few milliseconds later, the data on the P port changes, yet the $\overline{INT}$ signal remains high during the entire process. Why is this so?

Answer: The PCF8574 supports multiple data byte writes. Ensure that the master is not writing multiple data to the device. In the multiple data byte write mode, after the master sends the first byte, the slave acknowledges during the falling edge of the eighth data-bit clock cycle. On the rising edge of ACK clock cycle, the slave then updates the P port with the first data byte received from the master. The slave releases SDA on the falling edge of the ACK clock cycle, and the master sends the second data byte for writing to the slave. The slave acknowledges during the falling edge of the eighth data-bit clock cycle, and the slave updates the P port with the second data byte received from the master. The slave releases SDA on the falling edge of ACK clock cycle. The master may send a third (and more) data byte or a stop condition. However, although the data on the P port changed twice (or more) during the multiple-byte write cycles, $\overline{INT}$ remains high during the entire sequence.

Question 6: The master performs a read operation and reads 8-bit data from the PCF8574, but the data read is different from the eight bits on the I/O port of the PCF8574. Why is this so?

Answer: Before performing a read, the device outputs should be all high (either from power-up-reset initialization or from a write operation), so that an external driver can fully control the I/Os. If the output is written low before the read cycle, a low is read. If, for example, during the last write operation, the master wrote 00001111 to the P port of the PCF8574 and the external driver connected to the P port of the device sends 10101010 to the P port, upon performing a read operation, the master gets 00001010 as the data on the P port of the PCF8574.

References

1. Philips Semiconductors, *The I^{2}C Bus and How To Use It (Including Specifications)*, 98-8080-575-01, April 1995.
2. *MSP430F11x Mixed Signal Microcontroller,* data sheet, literature number SLAS256.
3. *PCA8550 Nonvolatile 5-Bit Register With I^{2}C Interface Technology and Applications,* application report, literature number SZZA014.
4. *PCF8574 Remote 8-Bit I/O Expander for I^{2}C Bus,* data sheet, literature number SCPS068.
5. *PCF8574A Remote 8-Bit I/O Expander for I^{2}C Bus,* data sheet, literature number SCPS069.
6. *THMC10 Remote/Local Temperature Monitor With SMBus Interface,* data sheet, literature number SLIS089.

Glossary

ACK	Acknowledge
$\overline{\text{ALERT}}$	Active low-temperature interrupt signal
CPU	Central processing unit
EMI	Electromagnetic interference
ESD	Electrostatic discharge
I^{2}C	Inter-integrated circuit, an industry-standard, two-wire, open-drain, communications protocol
IC	Integrated circuit
$\overline{\text{INT}}$	Interrupt output (active low)
I/O	Input/output
LED	Light-emitting diode
LSB	Least significant bit
Master	Device on the I^{2}C bus that controls the SCL line and initializes communication by sending a start condition, followed by an address word and R/$\overline{\text{W}}$ bit
MSB	Most significant bit
NACK	No acknowledge
PC	Personal computer
PCB	Printed circuit board
P port	Parallel (I/O) port
P read	Port read register
P write	Port write register
R/$\overline{\text{W}}$	Read/write bit
SCL	I^{2}C serial clock
SDA	I^{2}C serial data
Slave	Device on the I^{2}C bus that does not have control on the SCL line and is not responsible for initializing communication on the I^{2}C bus
SMBus	System-management bus

IMPORTANT NOTICE

Texas Instruments and its subsidiaries (TI) reserve the right to make changes to their products or to discontinue any product or service without notice, and advise customers to obtain the latest version of relevant information to verify, before placing orders, that information being relied on is current and complete. All products are sold subject to the terms and conditions of sale supplied at the time of order acknowledgment, including those pertaining to warranty, patent infringement, and limitation of liability.

TI warrants performance of its products to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are utilized to the extent TI deems necessary to support this warranty. Specific testing of all parameters of each device is not necessarily performed, except those mandated by government requirements.

Customers are responsible for their applications using TI components.

In order to minimize risks associated with the customer's applications, adequate design and operating safeguards must be provided by the customer to minimize inherent or procedural hazards.

TI assumes no liability for applications assistance or customer product design. TI does not warrant or represent that any license, either express or implied, is granted under any patent right, copyright, mask work right, or other intellectual property right of TI covering or relating to any combination, machine, or process in which such products or services might be or are used. TI's publication of information regarding any third party's products or services does not constitute TI's approval, license, warranty or endorsement thereof.

Reproduction of information in TI data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations and notices. Representation or reproduction of this information with alteration voids all warranties provided for an associated TI product or service, is an unfair and deceptive business practice, and TI is not responsible nor liable for any such use.

Resale of TI's products or services with *statements different from or beyond the parameters* stated by TI for that product or service voids all express and any implied warranties for the associated TI product or service, is an unfair and deceptive business practice, and TI is not responsible nor liable for any such use.

Also see: Standard Terms and Conditions of Sale for Semiconductor Products. www.ti.com/sc/docs/stdterms.htm

Mailing Address:

Texas Instruments
Post Office Box 655303
Dallas, Texas 75265