19-0729; Rev 2; 12/09

PECI-to-I²C Translator

General Description

The MAX6621 PECI-to-I²C translator provides an efficient, low-cost solution for PECI-to-SMBus™/I²C protocol conversion. The PECI-compliant host reads temperature data directly from up to four PECI-enabled CPUs. Interrupts are generated when the measured temperature exceeds the high-temperature limit and causes $\overline{\text{ALERT}}$ to assert. The $\overline{\text{RESET}}$ input allows the host to reset the I²C bus in the event of a communication error.

The I²C interface provides an independent serial communication channel to communicate synchronously with peripheral devices in a multiple master or multiple slave system. This interface allows a maximum serial-data rate of 400kbps.

The MAX6621 is designed to operate from a +3.0V to +3.6V supply voltage and ambient temperature range of -20°C to +120°C.

Features

- **400kbps I²C-Compatible, 2-Wire Serial Interface**
- **+3V to +3.6V Supply Voltage**
- **PECI-Compliant Port**
- **PECI-to-I²C Translation**
- **$\overline{\text{ALERT}}$ Output**
- **$\overline{\text{RESET}}$ Input (May Be Disabled as a Factory Option)**
- **Programmable Temperature Offsets**
- **-20°C to +120°C Operating Temperature Range**
- **V_{REF} Input Refers Logic Levels to the PECI Supply Voltage**
- **Automatic I²C Bus Lockup Timeout Reset**
- **Lead-Free, 10-Pin µMAX® Package**

Applications

Servers

Workstations

Desktop Computers

SMBus is a trademark of Intel Corp.

µMAX is a registered trademark of Maxim Integrated Products, Inc.

Ordering Information

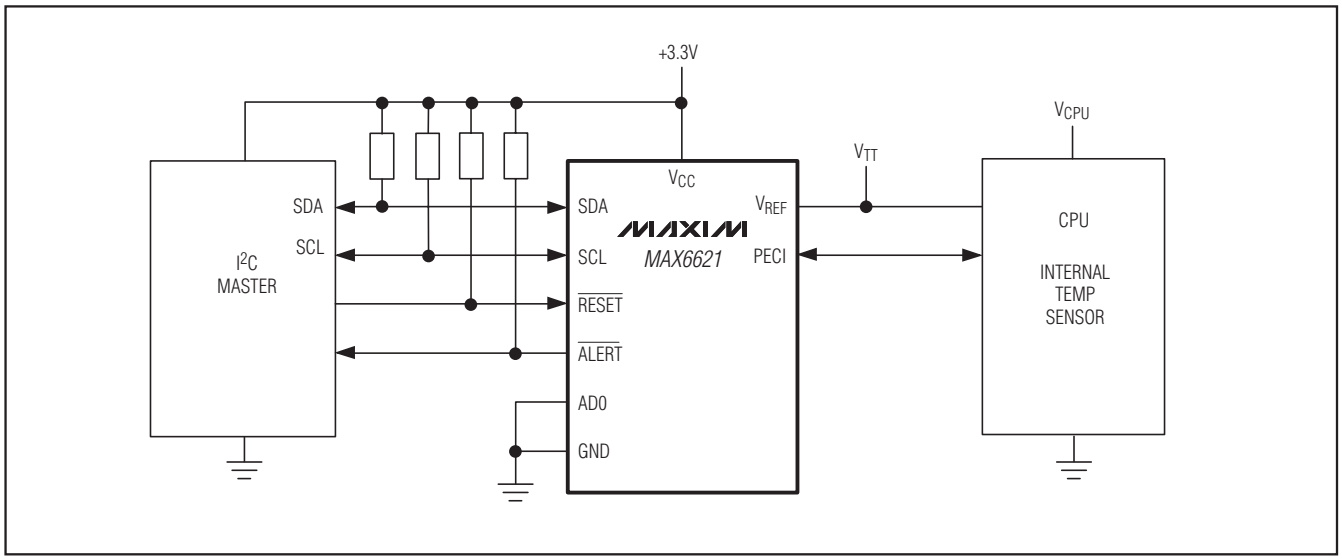
PART	TEMP RANGE	PIN-PACKAGE
MAX6621AUB+	-20°C to +120°C	10 µMAX
MAX6621AUB+T	-20°C to +120°C	10 µMAX

T = Tape and reel.

+Denotes a lead(Pb)-free/RoHS-compliant package.

Pin Configuration appears at end of data sheet.

Typical Application Circuit



For pricing, delivery, and ordering information, please contact Maxim Direct at 1-888-629-4642, or visit Maxim's website at www.maxim-ic.com.

ABSOLUTE MAXIMUM RATINGS

(All voltages with respect to GND.)

V_{CC} .....-0.3V to +4V
AD0, $\overline{RESET}$, $\overline{ALERT}$.....-0.3V to (V_{CC} + 0.3V)
SCL, SDA .....-0.3V to +6V
V_{REF} .....-0.3V to +4V
PECI .....-0.3V to (V_{REF} + 0.3V)
DC Current through SDA 10mA

Continuous Power Dissipation (T_A = +70°C)
10-Pin µMAX (derate 5.6mW/°C over T_A = +70°C).....444mW
Operating Temperature Range .....-20°C to +120°C
Junction Temperature.....+150°C
Storage Temperature Range .....-65°C to +150°C
Lead Temperature (soldering, 10s) .....+300°C

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 in the operational sections of the specifications is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

ELECTRICAL CHARACTERISTICS

(*Typical Application Circuit*, V_{CC} = +3V to +3.6V, V_{REF} = +0.95V to +1.26V, T_A = -20°C to +120°C, unless otherwise noted. Typical values are at V_{CC} = +3.3V, V_{REF} = +1.0V, T_A = +25°C.) (Note 1)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
SUPPLY						
Operating Supply Voltage	V_{CC}		3.0		3.6	V
Operating Supply Current	I_{CC}	SCL = 400kHz		4	7	mA
Power-On-Reset Voltage	V_{POR}		2.60		2.95	V
INPUT SCL, INPUT/OUTPUT SDA						
Low-Level Input Voltage	V_{IL}				0.3 x V_{CC}	V
High-Level Input Voltage	V_{IH}		0.7 x V_{CC}		5.5	V
Low-Level Output Voltage	V_{OL}	I_{OL} = 6mA			0.4	V
Leakage Current	I_L		-1		+1	µA
Input Capacitance	C_I			10		pF
$\overline{ALERT}$						
Low-Level Output Voltage	V_{OL}	I_{OL} = 6mA			0.4	V
ADDRESS INPUT AD0/$\overline{RST}$						
Low-Level Input Voltage	V_{IL}				0.3 x V_{CC}	V
High-Level Input Voltage	V_{IH}		0.7 x V_{CC}		V_{CC} + 0.3	V
Leakage Current	I_L		-2		+2	µA
Input Capacitance	C_I			10		pF
PECI						
Supply Voltage to PECI Cell	V_{REF}		0.95		1.26	V
Input Voltage Range	V_{IN}		-0.3		V_{REF} + 0.3	V
Low-Level Input Voltage Threshold	V_{IL}		0.275 x V_{REF}		0.500 x V_{REF}	V
High-Level Input Voltage Threshold	V_{IH}		0.550 x V_{REF}		0.725 x V_{REF}	V

ELECTRICAL CHARACTERISTICS (continued)

(*Typical Application Circuit*, V_{CC} = +3V to +3.6V, V_{REF} = +0.95V to +1.26V, T_A = -20°C to +120°C, unless otherwise noted. Typical values are at V_{CC} = +3.3V, V_{REF} = +1.0V, T_A = +25°C.) (Note 1)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
Hysteresis	V_H		0.1 x V_{REF}			V
Low-Level Sinking Current	I_{IL}		0.5		1.0	mA
High-Level Sourcing Current	I_{IH}		-6			mA
Input Capacitance	C_I	(Note 2)			10	pF
Signal-Noise Immunity Above 300MHz	V_N	(Note 2)	0.1 x V_{REF}			V_{P-P}

TIMING CHARACTERISTICS

(*Typical Application Circuit*, V_{CC} = +3V to +3.6V, V_{REF} = +0.95V to +1.26V, T_A = -20°C to +120°C, unless otherwise noted. Typical values are at V_{CC} = +3.3V, V_{REF} = +1.0V, T_A = +25°C.) (Note 2)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
$\overline{\text{RESET}}$ Pulse Width	$\overline{RST}$		100			ns
I²C INTERFACE						
Serial-Clock Frequency	f_{SCL}				400	kHz
Bus Free Time Between a STOP and a START Condition	t_{BUF}		1.3			µs
Hold Time, (Repeated) START Condition	$t_{HD, STA}$		0.6			µs
Repeated START Condition Setup Time	$t_{SU, STA}$		0.6			µs
STOP Condition Setup Time	$t_{SU, STO}$		0.6			µs
Data Hold Time	$t_{HD, DAT}$	(Note 3)			0.9	µs
Data Setup Time	$t_{SU, DAT}$		120			ns
SCL Clock-Low Period	t_{LOW}		1.3			µs
SCL Clock-High Period	t_{HIGH}		0.6			µs
Rise Time of Both SDA and SCL Signals, Receiving	t_R	(Notes 4, 5)		20 + 0.1C_b	300	ns
Fall Time of Both SDA and SCL Signals, Receiving	t_F	(Notes 4, 5)		20 + 0.1C_b	300	ns
Fall Time of SDA Transmitting	$t_{F.TX}$	(Notes 4, 5)		20 + 0.1C_b	250	ns
Pulse Width of Spike Suppressed	t_{SP}	(Note 6)	50	160		ns
Capacitive Load for Each Bus Line	C_b	(Note 4)			400	pF
PECI INTERFACE						
Bit Time (Note 7)	t_{BIT}	Overall time evident on PECI	0.495		500	µs
		Driven by MAX6621	0.495		250	

TIMING CHARACTERISTICS (continued)

(*Typical Application Circuit*, V_{CC} = +3V to +3.6V, V_{REF} = +0.95V to +1.26V, T_A = -20°C to +120°C, unless otherwise noted. Typical values are at V_{CC} = +3.3V, V_{REF} = +1.0V, T_A = +25°C.) (Note 2)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
Bit Time Jitter	$t_{BIT, jitter}$	Between adjacent bits in an PECI message header or data bytes after timing has been negotiated		1		%
Change in Bit Time	$t_{BIT, drift}$	Across a PECI address or PECI message bits as driven by MAX6621		2		%
High-Level Time for Logic-High	t_{H1}	(Note 8)	0.6	0.75	0.8	x t_{BIT}
High-Level Time for Logic-Low	t_{H0}		0.2	0.3	0.4	x t_{BIT}
Client Asserts PECI High During Logic-High	t_{SU}		0		0.2	x t_{BIT-M}
Rise Time	t_R	Measured from V_{OL} to V_PMAX, $V_{REF(nom)}$ -5% (Note 9)			30 + 5/Node	ns
Fall Time	t_F	Measured from V_{OH} to V_NMAX, $V_{REF(nom)}$ +5% (Note 9)			30/Node	ns
Hold Time	t_{HOLD}	Time for client to maintain a low idle drive after MAX6621 begins a message (Note 10)			0.5	x t_{BIT-1}
Stop Time	t_{STOP}	A constant low level driven by MAX6621 (Notes 8, 11)		2		x t_{BIT-M}
Maximum Dwell Time of the PECI Client	t_{RESET}	From the end of a ResetDevice command to the next message to which the reset client must be able to respond			0.4	ms
Minimum PECI Low Time Preceding a Message	t_{SETUP}	If the prior t_{BIT} is not known by MAX6621, the maximum t_{BIT} must be assumed and t_{SETUP} = 1ms in this case (Note 12)	2			x t_{BIT-X}

Note 1: All parameters are tested at T_A = +25°C. Specifications over temperature are guaranteed by design.

Note 2: Guaranteed by design; not production tested.

Note 3: A master device must provide a hold time of at least 300ns for the SDA signal (referred to V_{IL} of the SCL signal) to bridge the undefined region of SCL's falling edge.

Note 4: C_b = total capacitance of one bus line in pF. t_R and t_F measured between 0.3 x V_{CC} and 0.7 x V_{CC}.

Note 5: $I_{SINK} \le 6mA$. C_b = total capacitance of one bus line in pF. t_R and t_F measured between 0.3 x V_{CC} and 0.7 x V_{CC}.

Note 6: Input filters on the SDA and SCL inputs suppress noise spikes less than 50ns.

Note 7: The MAX6621 must drive a more restrictive time to allow for quantized sampling errors by a client yet still attain the minimum time less than 500µs. t_{BIT} limits apply equally to t_{BIT-A} and t_{BIT-M}.

Note 8: The minimum and maximum bit times are relative to t_{BIT} defined in the timing negotiation pulse.

Note 9: Extended trace lengths can appear as additional nodes.

Note 10: The client may deassert its low idle drive prior to the falling edge of the first bit of the message by using the rising edge to detect a message start. However, the time delay must be sufficient to qualify the rising edge as a true message rather than a noise spike.

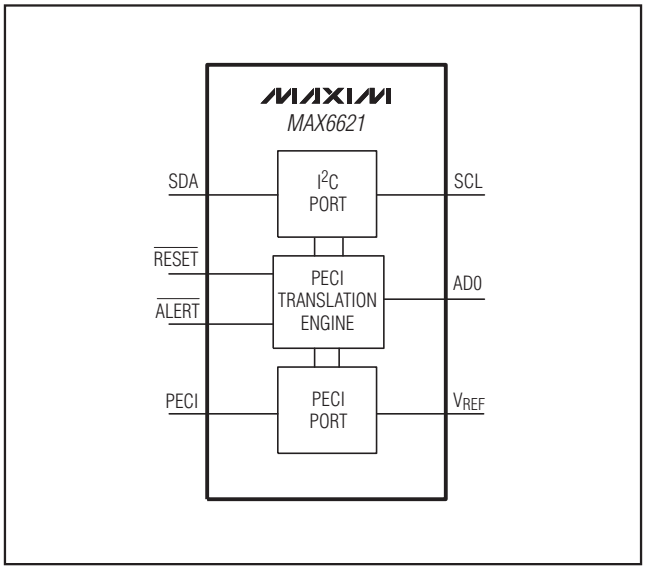
Note 11: The message stop is defined by two consecutive periods when the bus has no rising edge. Tolerance around this time is based on the t_{BIT-M} error budget.

Note 12: t_{SETUP} is not additive with t_{STOP}. Rather, these times may overlap.

Pin Description

PIN	NAME	FUNCTION
1	PECI	Platform Environment Control Interface (PECI) Serial-Bus Input/Output
2	AGND	Analog Ground
3	AD0	I²C Bus Device Address Selection Input
4	SDA	I²C Bus Data Input/Output
5	SCL	I²C Bus Clock Input/Output
6	V_{CC}	Power Supply. Bypass to GND with a 0.1µF capacitor.
7	GND	Power-Supply Ground
8	$\overline{RESET}$	I²C Reset Input. Pull $\overline{RESET}$ low to reset I²C interface and default all registers to startup values. Drive high for normal operation.
9	$\overline{ALERT}$	$\overline{ALERT}$ Interrupt Open-Drain Output. Asserts low when any temperature exceeds the programmed limit.
10	V_{REF}	PECI Input Supply Voltage. Bypass V_{REF} to AGND with a 0.1µF capacitor.

Block Diagram



Detailed Description

The MAX6621 obtains temperature data from an internal temperature sensor in PECI-compliant hosts. Up to four PECI hosts can be connected to the PECI I/O interface. The MAX6621 handles all the PECI transmissions and uses a 2-wire, I2C-compatible serial interface to communicate with the PECI host.

Registers and Commands

The following is an overview of the I2C/SMBus registers/commands supported by the MAX6621.

ADDRESS	DESCRIPTION	TRANSACTION TYPE
00h	Read socket 0, domain 0 temperature register	ReadWord
01h	Read socket 0, domain 1 temperature register	ReadWord
02h	Read socket 1, domain 0 temperature register	ReadWord
03h	Read socket 1, domain 1 temperature register	ReadWord
04h	Read socket 2, domain 0 temperature register	ReadWord
05h	Read socket 2, domain 1 temperature register	ReadWord
06h	Read socket 3, domain 0 temperature register	ReadWord
07h	Read socket 3, domain 1 temperature register	ReadWord
08h	Read maximum temperature for all enabled sockets/domains register	ReadWord
09h	Read firmware version register	ReadWord
0Ah	Read maximum temperature address	ReadWord
0Bh	Read socket and domain that caused alert	ReadWord
0Ch	Read/write CONFIG0 register	ReadWord/WriteWord
0Dh	Read/write CONFIG1 register	ReadWord/WriteWord
0Eh	Read/write CONFIG2 register	ReadWord/WriteWord
0Fh	Read/write CONFIG3 register	ReadWord/WriteWord
10h	Read/write alert temperature for socket 0	ReadWord/WriteWord
11h	Read/write alert temperature for socket 1	ReadWord/WriteWord
12h	Read/write alert temperature for socket 2	ReadWord/WriteWord
13h	Read/write alert temperature for socket 3	ReadWord/WriteWord
14h	Request polling	SendByte
15h	Clear alert	SendByte

Configuration

The MAX6621 has four configuration registers (Table 1). CONFIG0 is the main configuration register that enables the PECI sockets, I2C bus timeout, PEC, alert activation, and polling delay. CONFIG1 sets the number of retries, CONFIG2 sets the temperature offset, and CONFIG3 controls the temperature averaging. You can write to the configuration registers to set the configuration or read from the configuration registers to get the current settings.

Table 1. Configuration Registers

COMMAND BYTE	REGISTER DESCRIPTION	TYPE	RESULT
0Ch	CONFIG0 register	ReadWord/WriteWord	See the *CONFIG0* section.
0Dh	CONFIG1 register	ReadWord/WriteWord	See the *CONFIG1* section.
0Eh	CONFIG2 register	ReadWord/WriteWord	See the *CONFIG2* section.
0Fh	CONFIG3 register	ReadWord/WriteWord	See the *CONFIG3* section.

CONFIG0

The CONFIG0 register holds a bit mask for sockets and domains that are enabled for polling as well as a polling delay (minimum delay between sets of polls) and features enable/disable bits. Table 2 shows the various options for CONFIG0.

Table 2. CONFIG0 Register

BIT(S)	DESCRIPTION	DEFAULT
15:8	Polling enable for sockets and domains	00h
15	1 = enable socket 3, domain 1	0
14	1 = enable socket 3, domain 0	0
13	1 = enable socket 2, domain 1	0
12	1 = enable socket 2, domain 0	0
11	1 = enable socket 1, domain 1	0
10	1 = enable socket 1, domain 0	0
9	1 = enable socket 0, domain 1	0
8	1 = enable socket 0, domain 0	0
7	1 = enable I2C bus lockup timeout 0 = disable timeout	1
6	1 = alternate data representation 0 = 16-bit data representation	0
5	1 = enable I2C packet error checksum (PEC) on device return data 0 = disable PEC	1
4	1 = mask temperature alerts 0 = activate alerts	0
3	Reserved, set to 0	0
2:0	Poll delay, see Table 3	5

The optional polling delay (bits 2:0) inserts after polling the set of all sockets and domains that are enabled in bits 15:8 with a minimal pause of 2.5ms between PECI reads. After polling all enabled sockets and domains, the device pauses PECI communications for the configured time before starting to poll the set of enabled sockets and domains again. Table 3 shows the various polling delay options.

Table 3. Polling Delay

POLL DELAY VALUE	DELAY BETWEEN POLLS (ms)
0	Polling on request only
1	2.5
2	5
3	10
4	50
5	100 (default)
6	500
7	Reserved

CONFIG1

The CONFIG1 register configures the maximum number of retries before aborting a PECI temperature read as well as the originated (suggested) PECI bit time. Table 4 shows the various options for CONFIG1.

Table 4. CONFIG1 Register

BIT(S)	DESCRIPTION	DEFAULT
15:8	Originated PECI bit time (before negotiation) 01h: RESERVED 14h…0FFh: CONFIG1[15:8] + 1µs Minimum: 14h (= 21µs/47.62kHz) Maximum: 0FFh (= 256µs/3.906kHz)	02h
7:0	Maximum number of retries for PECI transactions	03h

CONFIG2

The CONFIG2 register holds the offset that is added to all temperature return values that are not error codes. The offset is enabled in CONFIG0, bit 6; +95°C is set as 17C0h or 005Fh, depending on the data format. To represent +95°C in 16-bit representation, convert +95°C to binary using two's complement and left-shift six times. The MAX6621 automatically converts the offset value to the equivalent value when the data format is changed. See Table 5 for the default offset and Table 6 for some example values.

Table 5. CONFIG2 Register

BIT(S)	DESCRIPTION	DEFAULT
15:0	Temperature offset	0000h

Table 6. Example Offset Values in 16-Bit Temperature Representation

TEMP (°C)	HEX	BINARY	
		RESHI	RESLO
0	0000h	0000 0000	0000 0000
+25	0640h	0000 0110	0100 0000
+50	0C80h	0000 1100	1000 0000
+75	12C0h	0001 0010	1100 0000
+95	17C0h	0001 0111	1100 0000

When configured in CONFIG2, and the return code is not an error code (see the *Error Codes* section), the device adds the offset value stored in CONFIG2 to the return value. For example, if the CPU's thermal control circuit activation point is at +95°C, CONFIG2 can be set to +95°C (005Fh or 17C0h) and all return values are converted to absolute temperatures. Note that the thermal control circuit activation point is CPU specific. The offset value is represented in the current data format.

CONFIG3

CONFIG3 register configures the temperature averaging function. See the *Temperature Averaging* section for more information. Table 7 shows the default settings.

Table 7. CONFIG3 Register

BIT(S)	DESCRIPTION	DEFAULT
15:8	Reserved, set to 0	00h
7:0	Averaging shift count, see formula	00h

Temperature Representation

Temperature data is formatted in 16-bit two's complement representing a range from -512°C to +512°C in steps of 1/64°C (Figure 1). Internally, the device always uses the 16-bit data format. The temperature is given in two's complement and left-shifted so that the +1°C bit is bit 6 (Figure 2). Temperatures can be represented externally in alternate data format if fractional readings are not needed. Table 8 shows some examples.

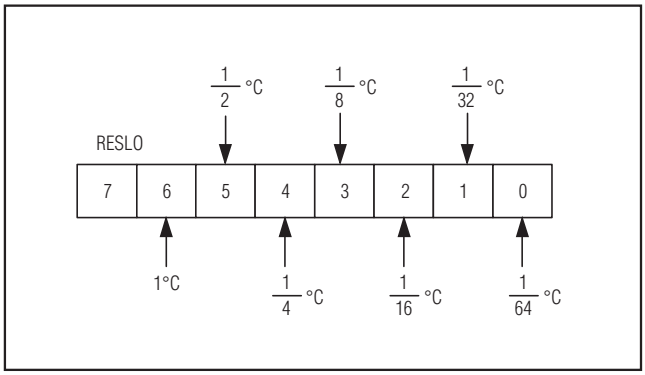


Figure 1. Temperature Measured in 1/64°C Steps

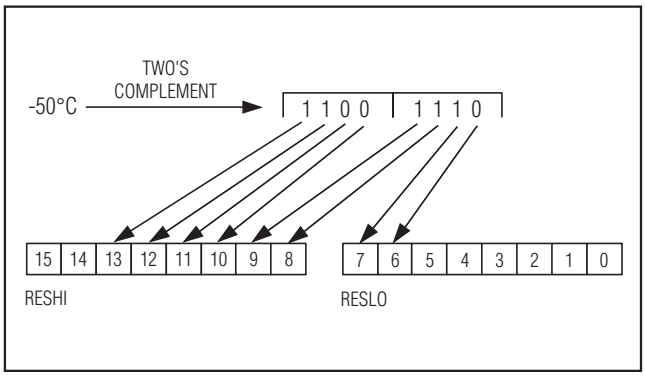


Figure 2. Conversion of Temperature Done in Two's Complement

Table 8. Example 16-Bit Representation with No Offset (Activation Point = +95°C)

TEMP (°C)	RELATIVE TEMP (°C)	HEX	BINARY	
			RESHI	RESLO
+94	-1	FFC0h	1111 1111	1100 0000
+85	-10	FD80h	1111 1101	1000 0000
+70	-25	FDC0h	1111 1101	1100 0000
+45	-50	F380h	1111 0011	1000 0000
+20	-75	ED30h	1110 1101	0100 0000

Alternate Temperature Value Representation

This optional feature can be enabled using bit 6 of CONFIG0. When the alternate data format is enabled, the temperature value is shifted right as shown in Table 9. The most significant bits are set to all 0s or all 1s depending on the sign bit 15, also shown as S in Figure 3. Table 10 shows some example values. This translation is not performed for error codes (16-bit values from 8000h through 81FFh).

Excluding error codes, the software only has to examine the RESLO data byte, as it represents an integer value in the range from -128°C to +127°C in 1°C steps. The RESHI byte is all 0s or all 1s for valid return codes, and either 80h or 81h for all error codes.

Temperature Averaging

The MAX6621 can average several temperature readings and return a value as calculated by:

$$T_{NEW} = \frac{1}{2^{CONFIG3}} \times T_{PECI} + \left(1 - \frac{1}{2^{CONFIG3}}\right) \times T_{OLD}$$

where T_{OLD} is the previously stored temperature, T_{PECI} is the new value read from PECI, and T_{NEW} is the newly stored temperature ready to be returned through I^2C. This calculation can cause significant bits to be lost.

Enable temperature averaging by writing the desired averaging amount to the CONFIG3 register. Writing 00h to the CONFIG3 register disables temperature averaging.

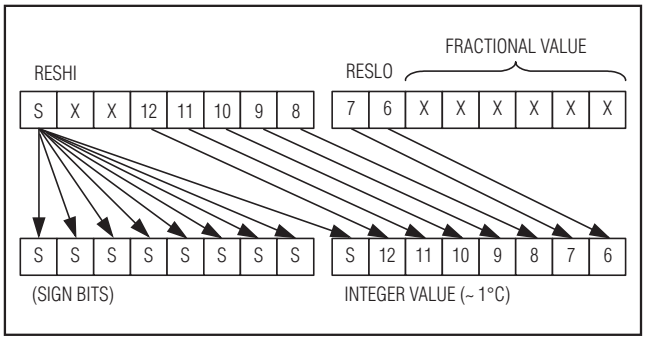


Figure 3. Alternate Temperature Representation

Table 9. Alternate Temperature Representation

DESCRIPTION	RESHI	RESLO
16-bit value	15:14:13:12:11:10:9:8	7:6:5:4:3:2:1:0
Alternate representation	15:15:15:15:15:15:15:15	15:12:11:10:9:8:7:6

Table 10. Example Alternate Representation with No Offset (Activation Point = +95°C)

TEMP (°C)	RELATIVE TEMP (°C)	HEX	BINARY	
			RESHI	RESLO
+94	-1	FFFFh	1111 1111	1111 1111
+85	-10	FFF6h	1111 1111	1111 0110
+70	-25	FFE7h	1111 1111	1110 0111
+45	-50	FFCEh	1111 1111	1100 1110
+20	-75	FFB5h	1111 1111	1011 0101

PECI-to-I²C Translator

Temperature Commands

Table 11 shows the different commands for selecting one of the PECI hosts or getting the maximum temperature. Read commands are initiated by the MAX6621, and the result returned is a 16-bit word with the least significant bit (LSB) clocked in first for the selected PECI host.

The result consists of RESLO for the 8 LSBs and RESHI for the 8 MSBs, resulting in a 16-bit word. The 16-bit words are temperature values read from the PECI interface. PECI-enabled Intel microprocessors return temperature data in fractions of 1°C below the thermal-control-circuit activation point, resulting in negative return values that do not represent absolute temperatures. Absolute temperatures can be achieved by setting the temperature offset in CONFIG2.

Table 12 shows example return values for an Intel CPU. Note that the MAX6621 does not interpret the return

Table 11. Read Temperature

ADDRESS	REGISTER	TYPE	RESULT
00h	Socket 0, domain 0	ReadWord	16-bit words
01h	Socket 0, domain 1		
02h	Socket 1, domain 0		
03h	Socket 1, domain 1		
04h	Socket 2, domain 0		
05h	Socket 2, domain 1		
06h	Socket 3, domain 0		
07h	Socket 3, domain 1		
08h	Read maximum temperature for all enabled sockets/domains		

Table 12. Return Temperature Values

RELATIVE TEMPERATURE (°C)	CONFIG2		RESHI:RESLO	
	16 BITS	ALTERNATE	16 BITS	ALTERNATE
-1	0000	0000	FFC0	FFFF
	17C0	005F	1780	005E
-36	0000	0000	F700	FFDC
	17C0	005F	0ec0	003B
-37	0000	0000	F6C0	FFDB
	17C0	005F	0E80	003A
-38	0000	0000	F680	FFDA
	17C0	005F	0E40	0039
-39	0000	0000	F640	FFD9
	17C0	005F	0E00	0038
-40	0000	0000	F600	FFD8
	17C0	005F	0DC0	0037
-41	0000	0000	F5C0	FFD7
	17C0	005F	0D80	0036
-42	0000	0000	F580	FFD6
	17C0	005F	0D40	0035
-43	0000	0000	F540	FFD5
	17C0	005F	0D00	0034

data (with the exception of error codes) and the relative temperatures are listed for reference only. Table 12 shows the values with 16-bit and alternate word format.

The read maximum temperature command from Table 11 returns the highest temperature that is not an error code from the enabled PECI sockets and domains. This operation works on signed numbers only and does not give information as to what socket the temperature result comes from. To find the socket and domain, use the read maximum temperature address command as shown in Table 13.

Table 13. Read Maximum Temperature Address

COMMAND	DESCRIPTION	TYPE	RESULT
0Ah	Read address of socket/domain with the maximum temperature	ReadWord	16-bit

The read maximum temperature address command returns the register that had the highest temperature when read maximum temperature was last called. An error is returned if the read maximum temperature has not been called or when the read maximum temperature itself returns an error.

Return Value Flow Chart

Figure 4 shows the operations performed on temperature data read through PECI.

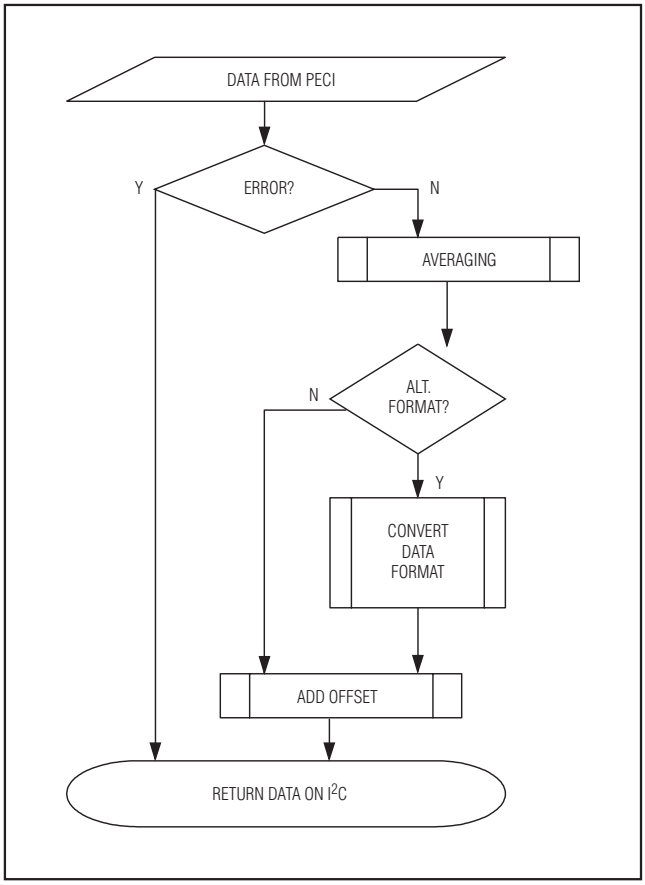


Figure 4. Operational Flowchart

Error Codes

Error codes are represented as 16-bit words in the 8000h–81FFh range as shown in Table 14.

Table 14. Error Codes

ERROR CODES	DESCRIPTION
8000h–80FFh	Refer to Intel PECI specification.
8100h	PECI transaction failed for more than the configured number of consecutive retries.
8101h	Polling disabled for requested socket/domain.
8102h	First poll not yet completed for requested socket/domain (on startup).
8103h	Read maximum temperature requested, but no sockets/domains enabled or all enabled sockets/domains have errors; or read maximum temperature address requested, but read maximum temperature was not called.
8104h	Get alert socket/domain requested, but no alert active.

$\overline{\text{ALERT}}$ Output

The MAX6621 asserts $\overline{\text{ALERT}}$ when a PECI temperature exceeds a configurable threshold after averaging. Table 15 below shows the registers that set the threshold and read the alert temperature for each socket. If there are no active alerts, an error is returned as shown in Table 14. If polling is disabled, the alert temperatures are only checked when the request polling command is called.

Once $\overline{\text{ALERT}}$ is asserted, a subsequent drop in temperature does not clear the alert. $\overline{\text{ALERT}}$ must be cleared by calling the clear alert command.

Table 15. Read/Write Temperature Alert Commands

COMMAND	DESCRIPTION	TYPE	DEFAULT
10h	Read/write alert temperature for socket 0	ReadWord/ WriteWord	7FFFh
11h	Read/write alert temperature for socket 1	ReadWord/ WriteWord	7FFFh
12h	Read/write alert temperature for socket 2	ReadWord/ WriteWord	7FFFh
13h	Read/write alert temperature for socket 3	ReadWord/ WriteWord	7FFFh

Clear Alert

The clear alert is shown in Table 16, and this command clears an active alert. If the temperature still exceeds one of the thresholds after clearing, $\overline{\text{ALERT}}$ reasserts on the next PECI temperature read. New alerts are disabled when mask alerts (CONFIG0, bit 4) is set. This does not affect an existing alert. The clear alert command needs to be called to clear it.

Table 16. Clear Alert

COMMAND	DESCRIPTION	TYPE
15h	Clear alert	SendByte

Read Alert Socket/Domain

Table 17 shows the read alert socket/domain command. Call this command to check for the socket/domain (register) that caused the alert to become active. If multiple temperatures exceed their respective thresholds, only the first violation is recorded. Register address is held in low byte.

Table 17. Read Alert Socket/Domain

COMMAND	DESCRIPTION	TYPE	RESULT
0Bh	Read alert socket and domain	ReadWord	16-bit word

The result is a 16-bit word (low byte transmitted first, high byte second) that contains the register that caused $\overline{ALERT}$ to assert. An error (8103h) is returned when there is no active $\overline{ALERT}$.

$\overline{RESET}$

The MAX6621 features a power-on reset (POR), bus lockout reset, and a reset input ($\overline{RESET}$). The power-on reset monitors V_{CC} and holds all outputs in high impedance until V_{CC} passes the POR threshold. The MAX6621 monitors V_{CC} for brownout conditions even after power-up.

Bus Lockout Timeout Reset

If an I2C transaction starts and gets locked up for greater than 20ms, the MAX6621 asserts the internal bus lockup reset that restarts itself in the default startup condition.

$\overline{RESET}$ Input

The MAX6621 features a $\overline{RESET}$ input that allows users to directly reset to the default startup conditions. Pull $\overline{RESET}$ low for a minimum of 10ns for a valid reset. The MAX6621 requires 100µs to be accessible after $\overline{RESET}$ has been asserted.

Version Information Command

Table 18 shows the command to read the firmware version.

Table 18. Firmware Command

COMMAND	DESCRIPTION	TYPE	RESULT
09h	Get firmware version	ReadWord	16 bit word

The result is a 16-bit word (low byte transmitted first, high byte second), e.g., 0100h for the MAX6621 firmware version 1.0.

Serial Interface

The MAX6621 operates as a slave that sends and receives data through an I2C-compatible, 2-wire interface. The interface uses a serial-data line (SDA) and a serial-clock line (SCL) to achieve bidirectional communication between master and slave. A master (typically a microcontroller) initiates all data transfers to and from the MAX6621 and generates the SCL clock that synchronizes the data transfer (Figure 5).

The MAX6621 SCL and SDA lines operate as both inputs and open-drain outputs. A pullup resistor is required on SCL and SDA.

Each transmission consists of a START condition sent by a master, followed by the MAX6621 7-bit slave address, plus an R/$\overline{W}$ bit, one or more data bytes, and finally a STOP condition (Figure 6). To write to a MAX6621 register, a write transmission consists of a START condition, followed by the MAX6621 7-bit slave address plus R/$\overline{W}$ = 0, a register address byte, one data byte, and finally a STOP condition. To read from a MAX6621 register, a combined write and read transmissions are required. The first write transmission consists of a START condition, followed by the MAX6621 7-bit slave address plus R/$\overline{W}$ = 0, a register address byte, and finally a STOP condition that sets the register to be read. The second read transmission consists of a START condition, followed by the MAX6621 7-bit slave address plus R/$\overline{W}$ = 1, one or more data bytes, and finally a STOP condition that reads the data from the

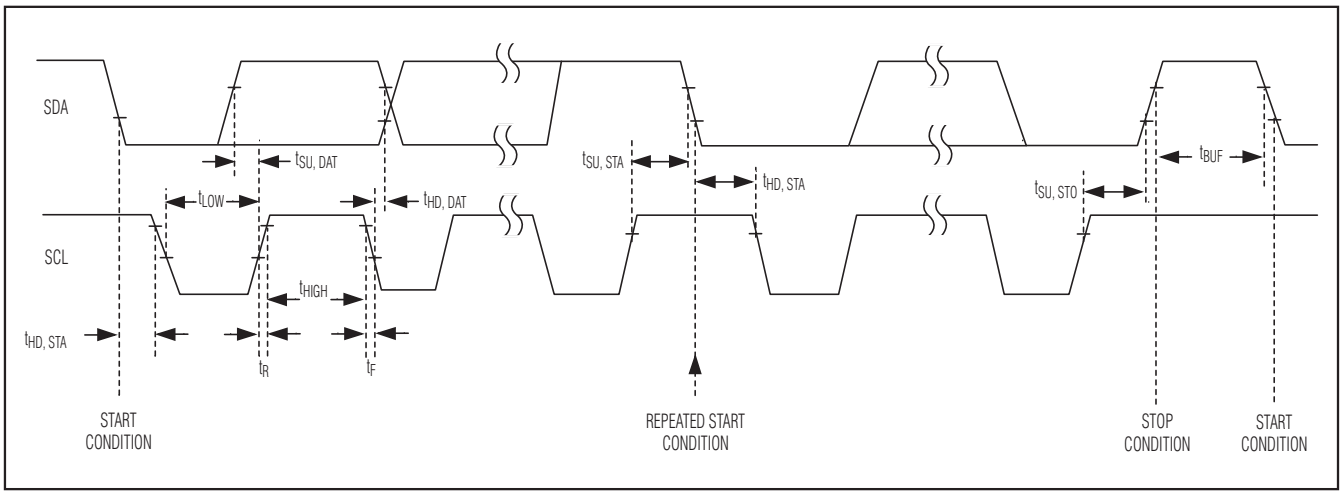


Figure 5. 2-Wire Serial-Interface Timing Details

specified register. These write and read transmissions can be joined using a repeated START even though the MAX6621 7-bit slave address needs to be present preceding the R/$\overline{W}$ bits.

Start and Stop Conditions

Both SCL and SDA remain high when the interface is not busy. A master signals the beginning of a transmission with a START (S) condition by transitioning SDA from high to low while SCL is high. When the master has finished communicating with the slave, it issues a STOP (P) condition by transitioning SDA from low to high while SCL is high. The bus is then free for another transmission (Figure 6).

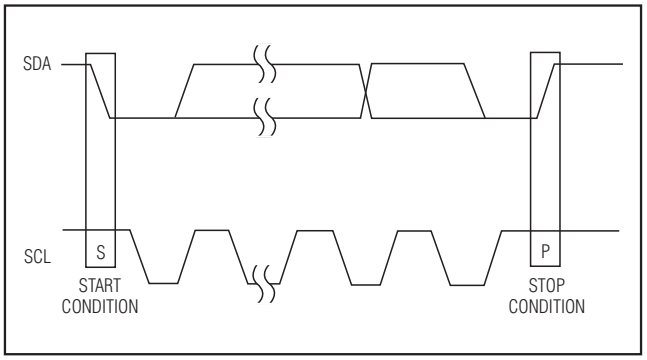


Figure 6. Start and Stop Conditions

Data Transfer and Acknowledge

One data bit is transferred during each clock pulse. The data on SDA must remain stable while SCL is high (Figure 7).

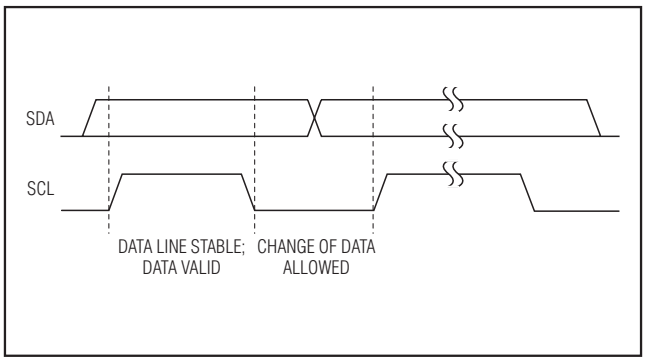


Figure 7. Bit Transfer

The acknowledge bit is a clocked 9th bit that the recipient uses to handshake receipt of each byte of data (Figure 8). Thus, each byte transferred effectively requires 9 bits. The master generates the 9th clock pulse, and the recipient pulls down SDA during the acknowledge clock pulse so that the SDA line is stable low during the high period of the clock pulse. When the master is transmitting to the MAX6621, the MAX6621 generates the acknowledge bit because the MAX6621 is the recipient. When the MAX6621 is transmitting to the master, the master generates the acknowledge bit because the master is the recipient.

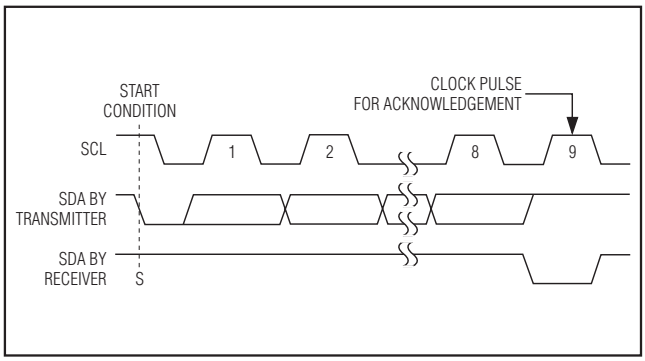


Figure 8. Acknowledge

Slave Address

The MAX6621 has a 7-bit long slave address (Figure 9). The 8th bit following the 7-bit slave address is the R/$\overline{W}$ bit. The R/$\overline{W}$ bit is low for a write command and high for a read command.

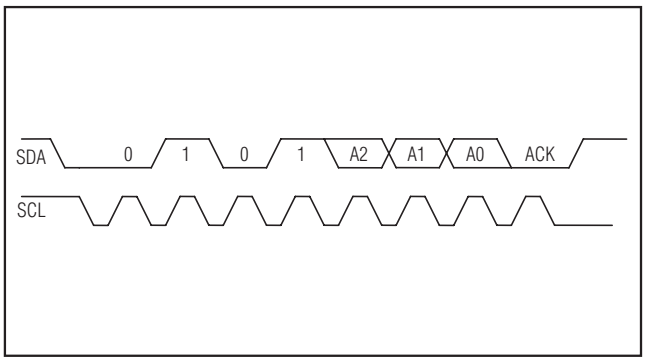


Figure 9. Slave Address

The first six bits of the MAX6621 slave address (A6:A1) are always 010101. A0 is selected by connecting AD0 to GND or V_{CC}. The resulting addresses are shown in Table 19. Other slave addresses may be possible as factory options, allowing multiple devices to be controlled independently from the same interface. Contact microcontroller technical support at **https://support.maxim-ic.com/micro**.

TYPICAL READ WORD COMMAND

PEC (PACKET ERROR CHECKSUM) ENABLED

MASTER	ADDR:7	W	A	CMD:8	A					
MAX6621	ADDR:7	R	A	RESLO:8	A	RESHI:8	A	PEC:8	NA	P

PEC (PACKET ERROR CHECKSUM) DISABLED

MASTER	ADDR:7	W	A	CMD:8	A			
MAX6621	ADDR:7	R	A	RESLO:8	A	RESHI:8	NA	P

TYPICAL WRITE WORD COMMAND

COMMAND WITH PEC (PACKET ERROR CHECKSUM)

MASTER	S	ADDR:7	W	A	CMD:8	A	INLO:8	A	INHI:8	A	PEC:8	A	P

COMMAND WITHOUT PEC (PACKET ERROR CHECKSUM)

MASTER	S	ADDR:7	W	A	CMD:8	A	INLO:8	A	INHI:8	A	P

THE RESULT CONSISTS OF RESLO FOR THE 8 LEAST SIGNIFICANT BITS (LSBS) AND RESHI FOR THE 8 MOST SIGNIFICANT BITS (MSBS), RESULTING IN A 16-BIT WORD. TEMPERATURE DATA AND ERROR CODES ARE GIVEN AS 16-BIT WORDS.

ADDR:7: 7-BIT ADDRESS FOLLOWED BY A READ (R = 1) OR WRITE (W = 0) BIT TO FORM THE 8-BIT ADDRESS USED IN THE I²C/SMBUS PROTOCOL.

P: I²C STOP CONDITION. SEE FIGURE 6.

S: I²C START CONDITION. SEE FIGURE 6.

A: ACK. THE PULSE ON THE 9th CLOCK CYCLE TO INDICATE ACKNOWLEDGE TRANSFER. SLAVE PULLS LOW TO GND AND MASTER PULLS TO SLAVE'S V_{OL}.

NA: NOT ACKNOWLEDGE

CMD: COMMAND BYTE

RESLO: LEAST SIGNIFICANT 8-BIT RESULT

RESHI: MOST SIGNIFICANT 8-BIT RESULT

Figure 10. Typical Read/Write Word Command

Table 19. MAX6621 Slave Addresses

A6:A1	A0 (SET BY AD0 PIN)	SLAVE ADDRESS	I2C ADDRESS BYTE INCLUDING R/W BIT
010101	0	2Ah	54h, 55h
010101	1	2Bh	56h, 57h

Message Format for Writing to the MAX6621

A write to the MAX6621 consists of the transmission of the MAX6621's slave address with the R/$\overline{W}$ bit set to zero, followed by at least 1 byte of information. The first byte of information is the command byte. The command byte determines which register of the MAX6621 is to be written to by the next byte or read from during the next read transmission. If a STOP condition is detected after the command byte is received, the MAX6621 takes no further action beyond setting the register address.

The bytes received after the command byte are data bytes. The data bytes go into the register of the MAX6621 specified by the command byte. Only the last data byte or word transmitted before a STOP condition is stored by the device (Figure 10).

Message Format for Reading the MAX6621

The MAX6621 is read using the MAX6621's internally stored command byte as an address pointer the same way the stored command byte is used as an address pointer for a write. The pointer autoincrements after each data byte is read. Thus, a read is initiated by first configuring the MAX6621's command byte by performing a write. The master can now read N consecutive bytes from the MAX6621 with the first data byte being read from the register addressed by the initialized command byte (Figure 10).

Packet Error Checksum (PEC)

All MAX6621 I2C packets have an optional packet error checksum (PEC). The PEC is implemented in accordance with the SMBus specification, versions 1.1 and 2. The MAX6621 accepts commands with or without PEC. The PEC for device responses is optional and can be disabled in the CONFIG0 register.

Applications Information

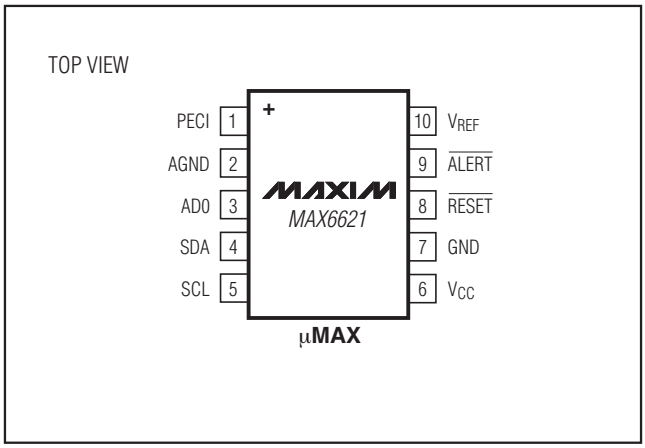
Operation with Multiple Masters

If the MAX6621 is operated on a 2-wire interface with multiple masters, a master reading the MAX6621 should use a repeated START between the write that sets the MAX6621's address pointer, and the read(s) that takes the data from the location(s) (Table 19). This is because it is possible for master 2 to take over the bus after master 1 has set up the MAX6621's address pointer, but before master 1 has read the data. If master 2 subsequently changes the MAX6621's address pointer, master 1's delayed read can be from an unexpected location. The use of multiple masters is not recommended.

Choosing Pullup Resistors

I2C requires pullup resistors to provide a logic-high level to data and clock lines. There are tradeoffs between power dissipation and speed, and a compromise must be made in choosing pullup resistor values. Every device connected to the bus introduces some capacitance even when the device is not in operation. I2C specifies a minimum 300ns rise time to go from low to high (30% to 70%) for fast mode, which is defined for a date rate of 400kbps (refer to the I2C specifications for details). To meet the rise time requirement, choose pullup resistors so that the rise time $t_R = 0.85R_{PULLUP} \times C_{BUS} < 300ns$. For typical low bus capacitances, a 4.7kΩ resistor can be used. For a bus capacitance of 400pF, choose a pullup resistor less than 880Ω. Many I2C devices work when the minimum specified rise time is not met. However, if the time it takes for the waveform to rise becomes too slow, these waveforms are not recognized by the master.

Pin Configuration



Chip Information

PROCESS: CMOS

Package Information

For the latest package outline information and land patterns, go to **www.maxim-ic.com/packages**. Note that a "+", "#", or "-" in the package code indicates RoHS status only. Package drawings may show a different suffix character, but the drawing pertains to the package regardless of RoHS status.

PACKAGE TYPE	PACKAGE CODE	DOCUMENT NO.
10 µMAX	U10+2	21-0061

PECI-to-I²C Translator

Revision History

REVISION NUMBER	REVISION DATE	DESCRIPTION	PAGES CHANGED
0	1/07	Initial release.	—
1	9/07	Changed the ADDRESS INPUT A0/RST# heading in the *Electrical Characteristics* table to ADDRESS INPUT AD0/RST#.	2
		Changed A0 to AD0 in the *Block Diagram*.	5
		Replaced Figure 9 and some text in the *Slave Address* section.	14, 15
		Replaced the text in the *I²C Address Range* section and replaced Table 19.	16
2	12/09	Corrected the equation for the originated PECI bit time (CONFIG1[15:8]) in Table 4.	7
		Corrected the *Slave Address* section.	15
		Clarified the slave addressing information in Table 19 and removed the *I²C Address Range* section.	16

Maxim cannot assume responsibility for use of any circuitry other than circuitry entirely embodied in a Maxim product. No circuit patent licenses are implied. Maxim reserves the right to change the circuitry and specifications without notice at any time.

© 2009 Maxim Integrated Products Maxim is a registered trademark of Maxim Integrated Products, Inc.