19-1942; Rev 3; 6/03

MAXIM

I²C-Compatible RTC in a TDFN

General Description

The MAX6900, I²C-bus-compatible real-time clock (RTC) in a 6-pin TDFN package contains a real-time clock/calendar and 31-byte × 8-bit wide of static random access memory (SRAM). The real-time clock/calendar provides seconds, minutes, hours, day, date, month, and year information. The end of the month date is automatically adjusted for months with fewer than 31 days, including corrections for leap year up to the year 2100. The clock operates in either the 24hr or 12hr format with an AM/PM indicator.

Applications

Portable Instruments

Point-of-Sale Equipment

Intelligent Instruments

Battery-Powered Products

Features

- **Real-Time Clock Counts Seconds, Minutes, Hours, Date, Month, Day, and Year**
- **Leap Year Compensation Valid up to Year 2100**
- **Fast (400kHz) I²C-Bus-Compatible Interface from 2.0V to 5.5V**
- **31 × 8 SRAM for Scratchpad Data Storage**
- **Uses Standard 32.768kHz, 12.5pF Load, Watch Crystal**
- **Ultra-Low 225nA (typ) Timekeeping Current**
- **Single-Byte or Multiple-Byte (Burst Mode) Data Transfer for Read or Write of Clock Registers or SRAM**
- **6-Pin 3mm x 3mm x 0.8mm TDFN Surface-Mount Package**
- **No External Crystal Bias Resistors or Capacitors Required**

Ordering Information

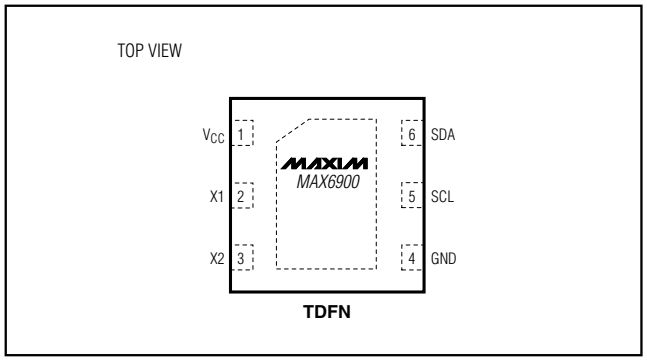
PART	TEMP RANGE	PIN-PACKAGE	TOP MARK
MAX6900ETT-T	-40°C to +85°C	6 TDFN	AEU

Related Real-Time Clock Products

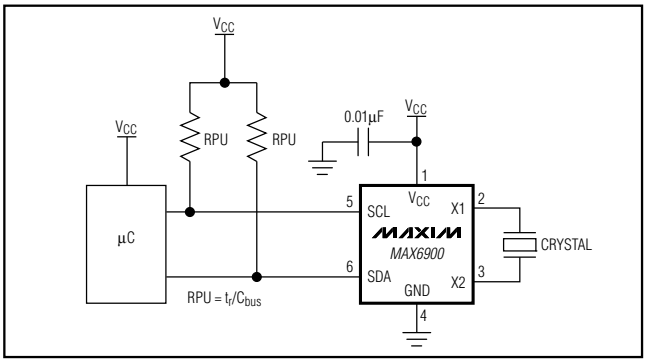
PART	SERIAL BUS	SRAM	ALARM FUNCTION	OUTPUT FREQUENCY	PIN-PACKAGE
MAX6900	I²C compatible	31 × 8	—	—	6 TDFN
MAX6901	3-wire	31 × 8	Polled	32kHz	8 TDFN
MAX6902	SPI™ compatible	31 × 8	Polled	—	8 TDFN

SPI is a trademark of Motorola, Inc.

Pin Configuration



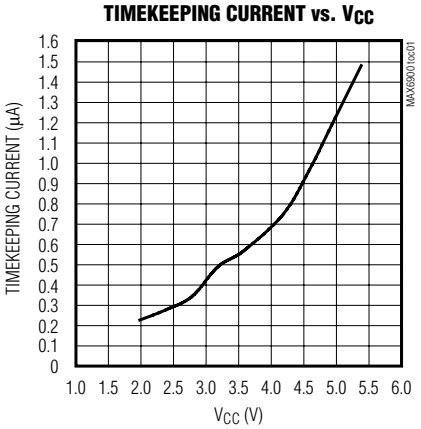
Typical Operating Circuit



I²C-Compatible RTC in a TDFN

Typical Operating Characteristics

(T_A = +25°C, unless otherwise noted.)



Pin Description

PIN	NAME	FUNCTION
1	V_{CC}	Power Supply
2	X1	32.768kHz External Crystal
3	X2	32.768kHz External Crystal
4	GND	Ground
5	SCL	I²C-Bus-Compatible Clock Input
6	SDA	I²C-Bus-Compatible Data Input/Output
—	PAD	Ground

Detailed Description

The MAX6900 contains eight timekeeping registers, burst address registers, a control register, an on-chip 32.768kHz oscillator circuit, and a serial 2-wire, I²C-compatible interface. There are also 31 bytes, 8 bits wide of SRAM on board. Time and calendar data are stored in the registers in a binary-coded decimal (BCD) format. Figure 1 shows an I²C-bus-compatible timing diagram. Figure 2 shows the MAX6900 functional diagram.

Real-Time Clock

The RTC provides seconds, minutes, hours, day, date, month, and year information. The end of the month is automatically adjusted for months with fewer than 31

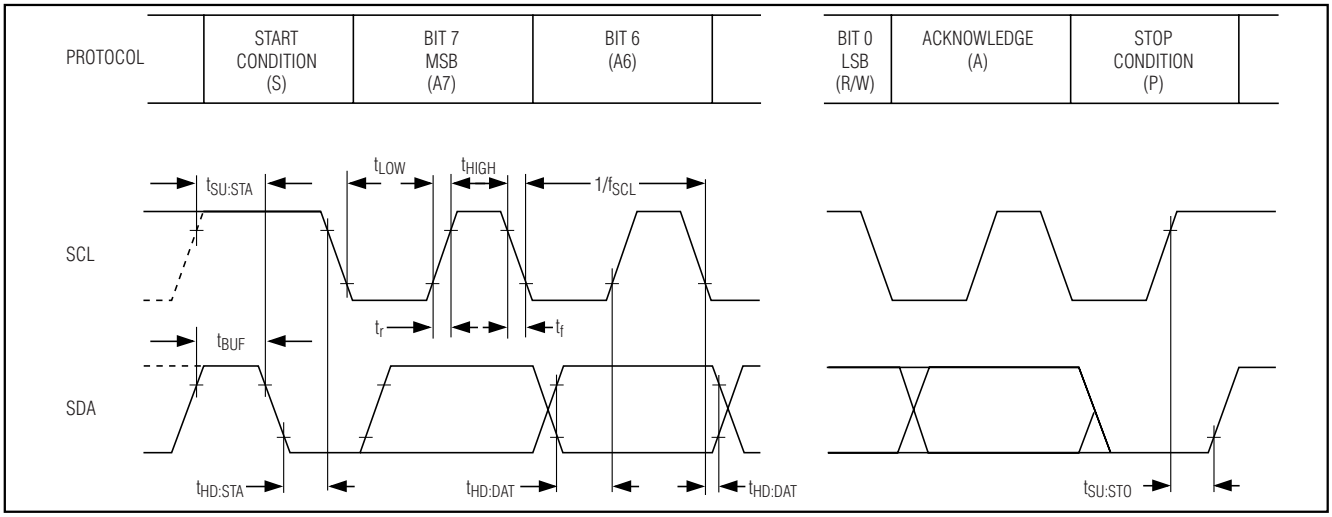


Figure 1. Detailed I²C-Bus Timing Diagrams

I²C-Compatible RTC in a TDFN

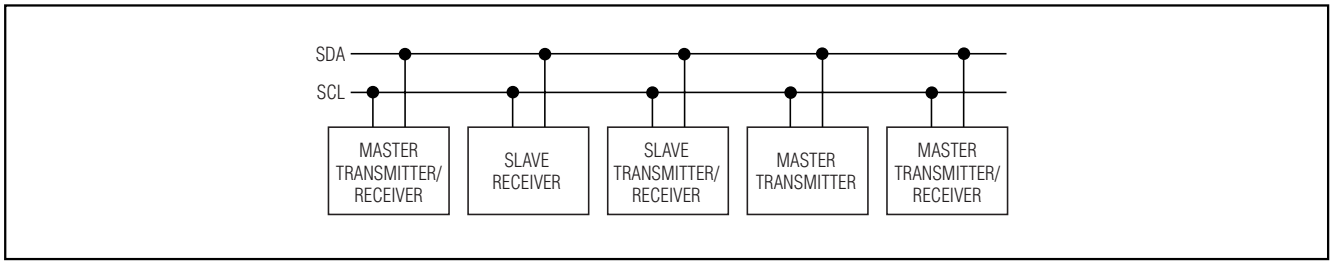


Figure 3. I²C Bus System Configuration

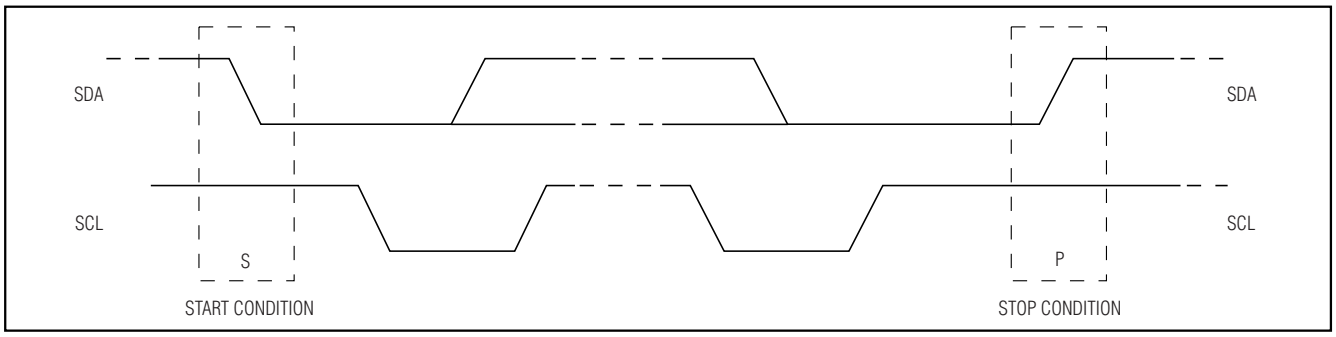


Figure 4. I²C Bus Start and Stop Conditions

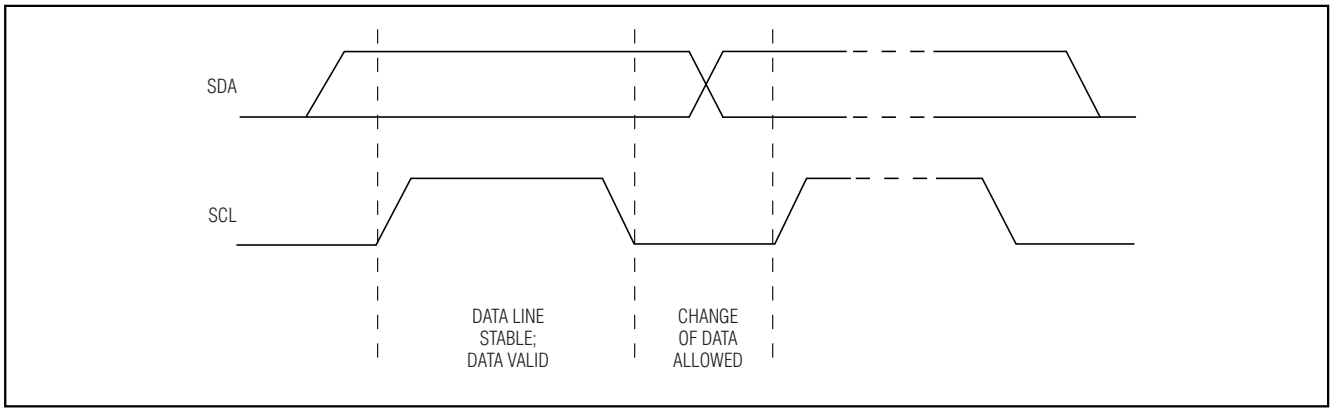


Figure 5. I²C Bus Bit Transfer

After the Start condition occurs, 1 bit of data is transferred for each clock pulse. The data on SDA must remain stable during the high portion of the clock pulse as changes in data during this time are interpreted as a control signal (Figure 5). Any time a start condition occurs, the Slave ID must follow immediately, regardless of completion of the previous data transfer.

Before any data is transmitted on the I²C-bus-compatible serial interface, the device that is expected to respond is addressed first. The first byte sent after the start (S) procedure is the Address byte or 7-bit Slave ID. The MAX6900 acts as a slave transmitter/receiver. Therefore, SCL is only an input clock signal and SDA is a bidirectional data line. The Slave Address for the MAX6900 is shown in Figure 6.

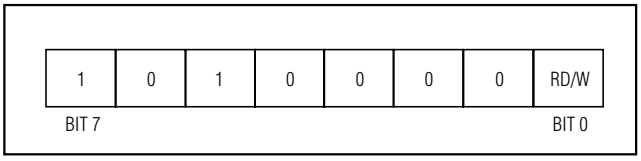


Figure 6. I²C Bus Slave Address or 7-Bit Slave ID

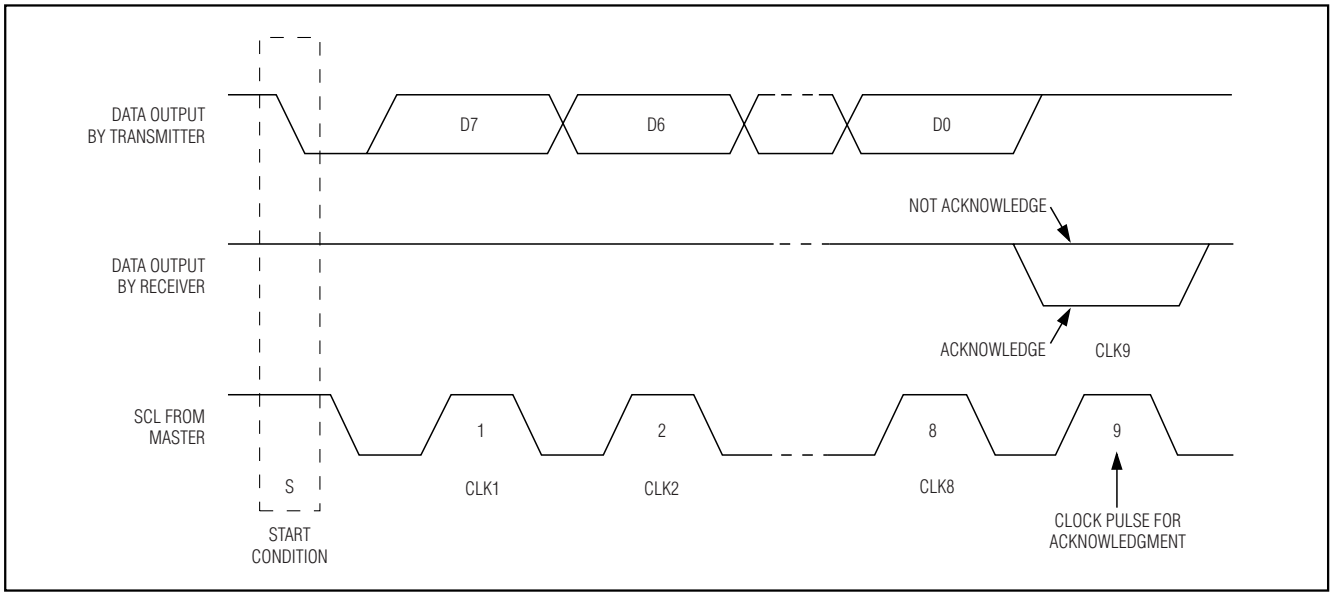


Figure 7. I²C Bus Acknowledge

An unlimited number of data bytes between the start and stop conditions can be sent between the transmitter and receiver. Each 8-bit byte is followed by an acknowledge bit. Also, a master receiver must generate an acknowledge after each byte it receives that has been clocked out of the slave transmitter.

The device that acknowledges must pull down the SDA line during the acknowledge clock pulse (Figure 7), so that the SDA line is stable low during the high period of the acknowledge clock pulse (setup and hold times must also be met). A master receiver must signal an end of data to the transmitter by not generating an acknowledge on the last byte that has been clocked out of the slave. In this case, the transmitter must leave SDA high to enable the master to generate a stop condition. Any time a stop condition is received before the current byte of data transfer is complete, the last incomplete byte is ignored.

The second byte of data sent after the start condition is the Address/Command byte (Figure 8). Each data transfer is initiated by an Address/Command byte. The MSB (bit 7) must be a logic 1. When the MSB is zero, Writes to the MAX6900 are disabled. Bit 6 specifies clock/calendar data if logic 0 or RAM data if logic 1 (Tables 1 and 2). Bits 1 through 5 specify the designated registers to be input or output. The LSB (bit 0) specifies a Write operation (input) if logic 0 or Read operation (output) if logic 1. The Command byte is always input starting with the MSB (bit 7).

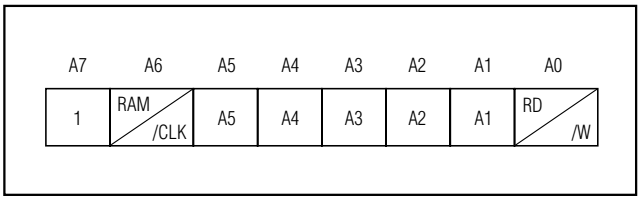


Figure 8. Address/Command Byte

Reading from the Timekeeping Registers

The timekeeping registers (Seconds, Minutes, Hours, Date, Month, Day, Year, and Century) read either with a Single Read or a Burst Read. Since the clock runs continuously and a Read takes a finite amount of time, it is possible that the clock counters could change during a Read operation, thereby reporting inaccurate timekeeping data. In the MAX6900, the clock counter data is buffered by a latch. Clock counter data is latched by the I²C-bus-compatible read command (on the falling edge of SCL when the Slave Acknowledge bit is sent after the Address/Command byte has been sent by the master to read a timekeeping register). Collision-detection circuitry ensures that this does not happen coincident with a seconds counter update to ensure accurate time data is being read. This avoids time data changes during a Read operation. The clock counters continue to count and keep accurate time during the Read operation.

When using a Single Read to read each of the timekeeping registers individually, perform error checking

I²C-Compatible RTC in a TDFN

Table 1. Register Address Definition

FUNCTION CLOCK	A7	A6	A5	A4	A3	A2	A1	A0		VALUE	D7	D6	D5	D4	D3	D2	D1	D0
SEC	**1**	**0**	**0**	**0**	**0**	**0**	**0**	RD		00-59	7	10 SEC			1 SEC			
								/W		***POR STATE**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
MIN	**1**	**0**	**0**	**0**	**0**	**0**	**1**	RD		00-59	0	10 MIN			1 MIN			
								/W		***POR STATE**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
HR	**1**	**0**	**0**	**0**	**0**	**1**	**0**	RD		00-23	12/24	0	10 HR	10 HR	1 HR			
								/W		01-12	1/0		A/P 0/1					
										***POR STATE**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
DATE	**1**	**0**	**0**	**0**	**0**	**1**	**1**	RD		01-28/29 01-30 01-31	0	0	10 DATE		1 DATE			
								/W		***POR STATE**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**1**
MONTH	**1**	**0**	**0**	**0**	**1**	**0**	**0**	RD		01-12	0	0	0	10M	1 MONTH			
								/W		***POR STATE**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**1**
DAY	**1**	**0**	**0**	**0**	**1**	**0**	**1**	RD		01-07	0	0	0	0	0	WEEK DAY		
								/W		***POR STATE**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**1**
YEAR	**1**	**0**	**0**	**0**	**1**	**1**	**0**	RD		00-99	10 YEAR				1 YEAR			
								/W		***POR STATE**	**0**	**1**	**1**	**1**	**0**	**0**	**0**	**0**
CONTROL	**1**	**0**	**0**	**0**	**1**	**1**	**1**	RD			WP	0	0	0	0	0	0	0
								/W		***POR STATE**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
CENTURY	**1**	**0**	**0**	**1**	**0**	**0**	**1**	RD		00-99	1000 YEAR				100 YEAR			
								/W		***POR STATE**	**0**	**0**	**0**	**1**	**1**	**0**	**0**	**1**
RESERVED	1	0	0	1	0	1	1	RD			0	0	0	0	0	0	0	0
								/W		***POR STATE**	0	0	0	0	0	1	1	1
CLOCK BURST	**1**	**0**	**1**	**1**	**1**	**1**	**1**	RD										
								/W										

Table 1. Register Address Definition (continued)

FUNCTION CLOCK	A7	A6	A5	A4	A3	A2	A1	A0	VALUE	D7	D6	D5	D4	D3	D2	D1	D0
RAM																	
RAM 0	**1**	**1**	**0**	**0**	**0**	**0**	**0**	RD /W	**RAM DATA 0**	x	x	x	x	x	x	x	x
RAM 30	**1**	**1**	**1**	**1**	**1**	**1**	**0**	RD /W	**RAM DATA 30**	x	x	x	x	x	x	x	x
RAM BURST	**1**	**1**	**1**	**1**	**1**	**1**	**1**	RD /W									

Note: POR STATE *defines power-on reset state of register contents.*

Table 2. Hex Register Address Definition

HEX REGISTER ADDRESS/DESCRIPTION			
WRITE ADDRESS/ COMMAND BYTE (HEX)	**READ ADDRESS/ COMMAND BYTE (HEX)**	**DESCRIPTION**	**POR CONTENTS**
80	81	Seconds	00
82	83	Minutes	00
84	85	Hours	00
86	87	Date	01
88	89	Month	01
8A	8B	Day	01
8C	8D	Year	70
8E	8F	Control	00
92	93	Century	19
96	97	Reserved	07
BE	BF	Clock Burst	N/A
C0	C1	RAM 0	Indeterminate
C2	C3	RAM 1	Indeterminate
C4	C5	RAM 2	Indeterminate
C6	C7	RAM 3	Indeterminate
C8	C9	RAM 4	Indeterminate
CA	CB	RAM 5	Indeterminate
CC	CD	RAM 6	Indeterminate
CE	CF	RAM 7	Indeterminate

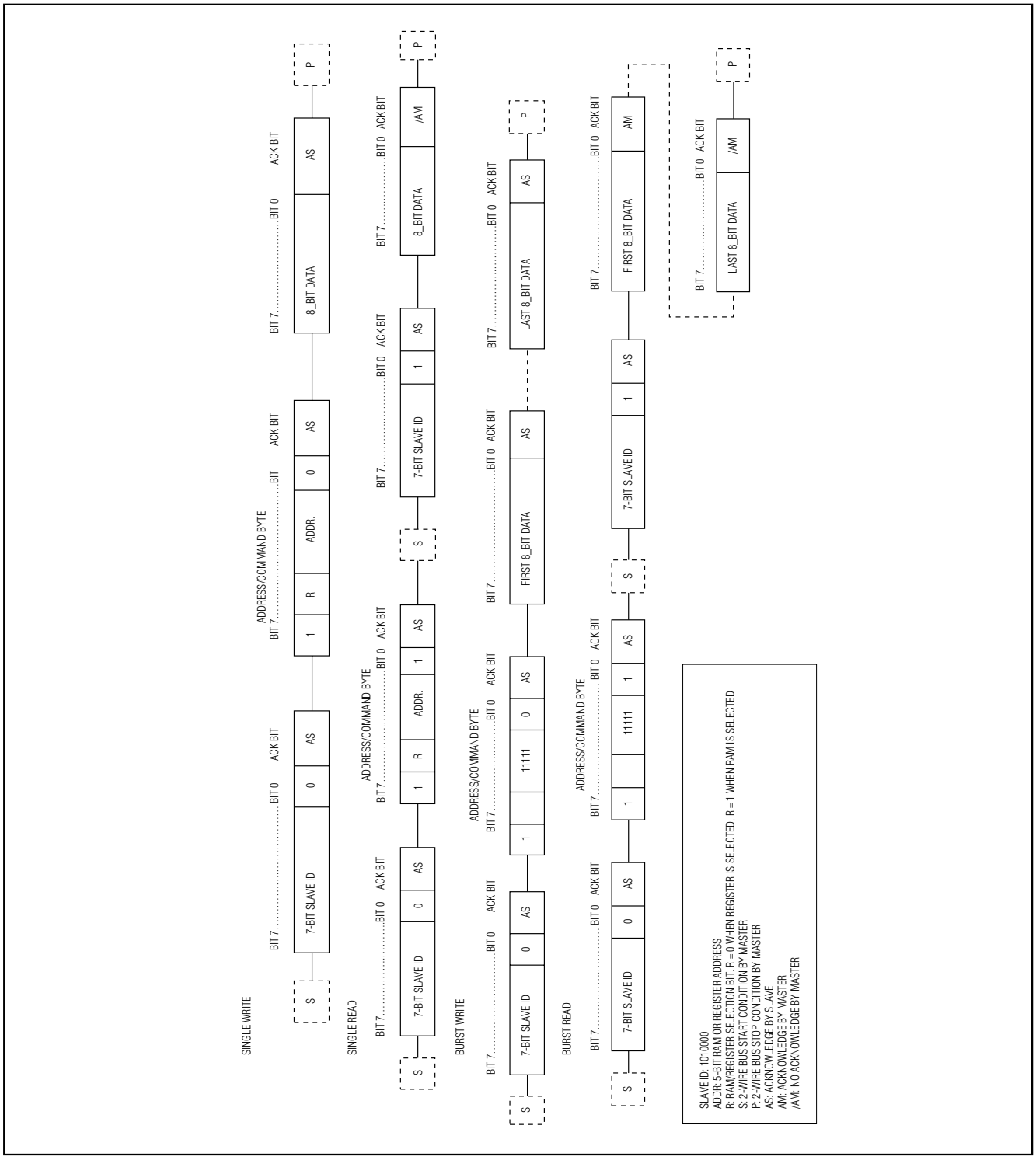
Table 2. Hex Register Address Definition (continued)

HEX REGISTER ADDRESS/DESCRIPTION			
WRITE ADDRESS/ COMMAND BYTE (HEX)	READ ADDRESS/ COMMAND BYTE (HEX)	DESCRIPTION	POR CONTENTS
D0	D1	RAM 8	Indeterminate
D2	D3	RAM 9	Indeterminate
D4	D5	RAM 10	Indeterminate
D6	D7	RAM 11	Indeterminate
D8	D9	RAM 12	Indeterminate
DA	DB	RAM 13	Indeterminate
DC	DD	RAM 14	Indeterminate
DE	DF	RAM 15	Indeterminate
E0	E1	RAM 16	Indeterminate
E2	E3	RAM 17	Indeterminate
E4	E5	RAM 18	Indeterminate
E6	E7	RAM 19	Indeterminate
E8	E9	RAM 20	Indeterminate
EA	EB	RAM 21	Indeterminate
EC	ED	RAM 22	Indeterminate
EE	EF	RAM 23	Indeterminate
F0	F1	RAM 24	Indeterminate
F2	F3	RAM 25	Indeterminate
F4	F5	RAM 26	Indeterminate
F6	F7	RAM 27	Indeterminate
F8	F9	RAM 28	Indeterminate
FA	FB	RAM 29	Indeterminate
FC	FD	RAM 30	Indeterminate
FE	FF	RAM Burst	N/A

on the receiving end. The potential for errors occurs when the seconds counter increments before all the other registers are read out. For example, suppose a carry of 13:59:59 to 14:00:00 occurs during Single Read operations of the timekeeping registers. The net data could become 14:59:59, which is erroneous real-time data. To prevent this with Single Read operations, read the Seconds register first (initial seconds) and store this value for future comparison. When the remaining timekeeping registers have been read out, read the Seconds register again (final seconds). If the initial seconds value is 59, check that the final seconds value is still 59; if not, repeat the entire Single Read process for the timekeeping registers. A comparison of the initial seconds value with the final seconds value can indicate if there was a bus delay problem in reading the timekeeping data (difference should always be 1s or less). Using a 100kHz bus speed, sequential Single Reads take under 2.5ms to read all seven of the timekeeping registers plus a second read of the Seconds register.

The most accurate way to read the timekeeping registers is a Burst Read. In the Burst Read mode, the main timekeeping registers (Seconds, Minutes, Hours, Date, Month, Day, Year) and the Control register are read sequentially. All of the main timekeeping registers and the Control register must be read out as a group of eight registers, with 8 bytes each, for proper execution

Table 3. Data Transfer Summary



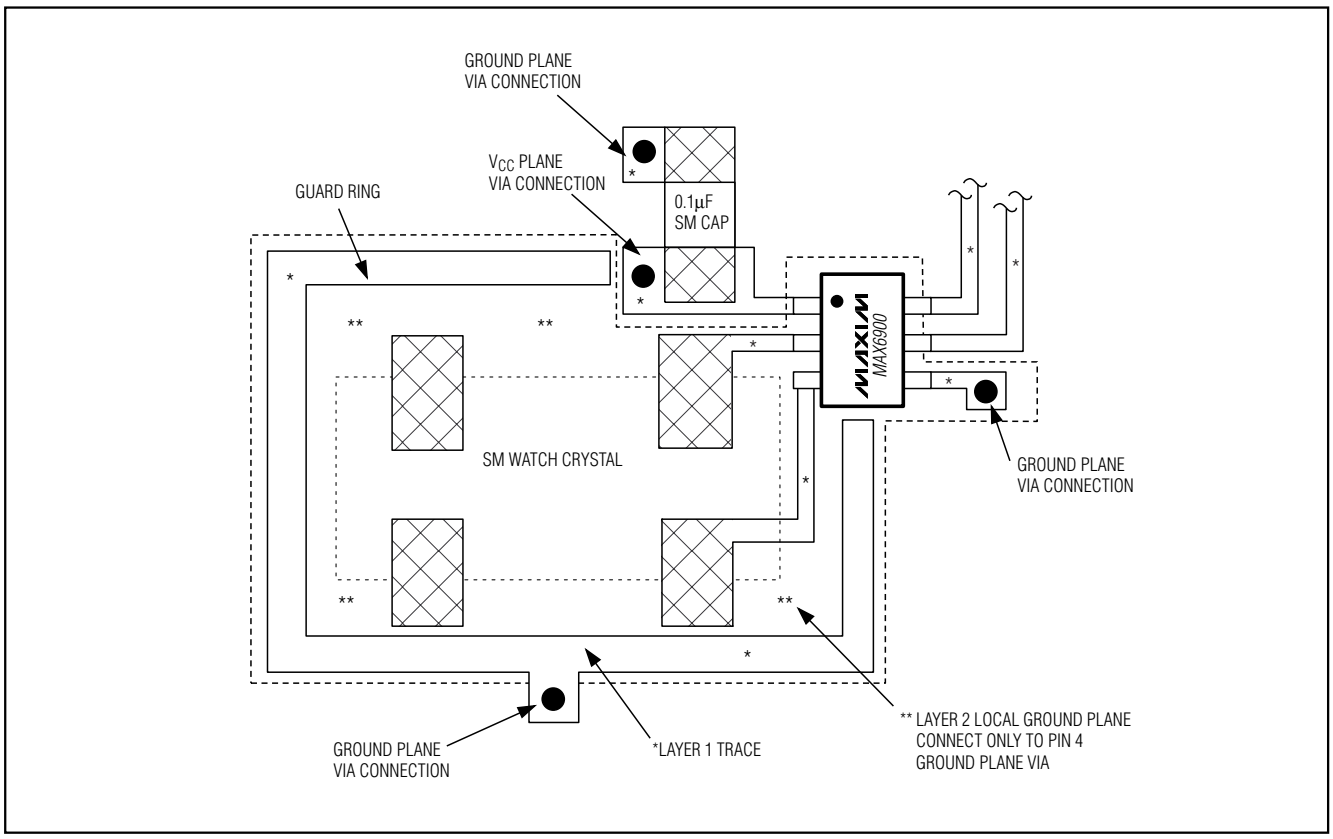


Figure 11. Recommended Board Layout

I²C-Compatible RTC in a TDFN

Package Information

(The package drawing(s) in this data sheet may not reflect the most current specifications. For the latest package outline information go to **www.maxim-ic.com/packages**.)