X1205
实时时钟／日历

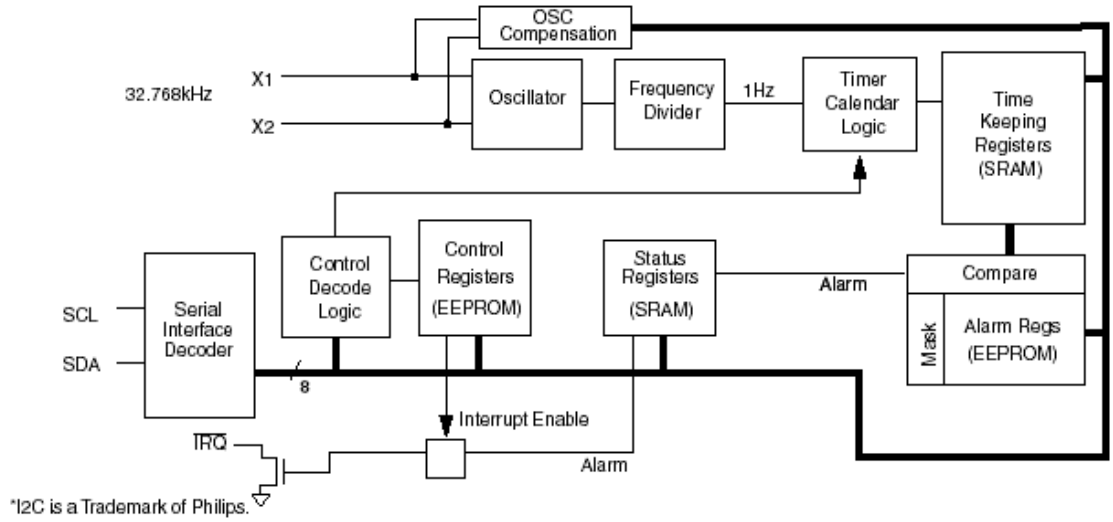
特性

- 实时时钟／日历
  －按小时，分钟和秒追踪时间
  －星期、日、月、年
- 双路报警（非易失性）
  —可设置月，日，星期，时，分，秒
  —重置方式（周期中断）
- 片内的振荡器补偿
  －内含反馈电阻和补偿电容
- －64个位置数控调整电容
  －6个数字频率调节设置，可设置的范围是±30ppm
- 电池切换或大电容输入
- 兼容于I^2C的两线^{TM}接口
  －400KHz的数据传输速率
- 低功耗CMOS
  －1.25μA工作电流(典型值)
- 小型封装
  －8引脚SOIC和8引脚TSSOP

应用

- 多功能电表
- HAVC 设备
- 音频/视频元件
- 机顶盒/电视机
- 调制解调器
- 网络路由器，集线器,开关，桥式电路
- 蜂窝结构设备
- 无线宽带安装设备
- 寻呼机/PDA
- POS设备
- 测试仪表/装置
- 办公自动化（复印机，传真机）
- 家用仪器
- 计算机设备
- 其它工用/医用/汽车工业设备

方框图



描述

X1205 是一个带有时钟、日历、两路报警、振荡器补偿和电池切换的实时时钟。

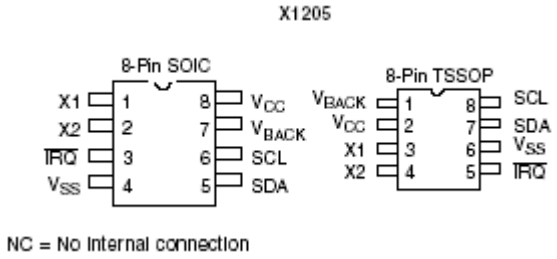
振荡器用一个外部的、低价格的 32.768Khz 晶体。所有补偿和调整元件集成于芯片上。这样除去了外部的离散元件和一个调整电容，节约电路板空间和元器件的费用。

实时时钟用分别的时、分、秒寄存器跟踪时间。日历有分别的日期、星期、月和年寄存器，日历可正确通过 2099 年，具有自动闰年修正功能。

强大的双报警功能，能够被设置到任何时钟／日历值上，与报警相匹配。例如，每分钟，每个星期二或三月21日上午5:23均可。报警能够在状态寄存器中被查询或提供一个硬件的中断（$\overline{IRQ}$管脚），这是一个重复模式，报警容许产生一个周期性的中断。

该器件提供一个备份电源输入脚V_{BACK}，该脚容许器件用电池或大容量电容进行备份供电。整个X1205器件的工作电压范围为2.7 V至5.5V，X1205的时钟／日历部分的工作可降到1.8V(待机模式)。

引脚排列图



NC = No Internal connection

串行时钟 (SCL)

SCL输入端被用来作为数据输入和输出的时钟同步信号，该输入端的输入缓冲器总是激活的。

串行数据 (SDA)

SDA是一个双向引脚，用于对器件输入或输出数据。它是一个漏极开路输出，可以与其它漏极开路或集电极开路输出端线或。输入缓冲器总是激活的。

漏极开路输出需要使用上拉电阻。输出电路使用一个斜率控制的下拉来控制输出信号的下降时间。电路被设计成400kHz速率的2线接口。

后备电压V_{BACK}

这个输入端为器件提供一个备用电源电压。当V_{CC}电源失效时V_{BACK}为器件提供电源。可将该引脚连接到电池、大容量电容，如果不用应将其接地。

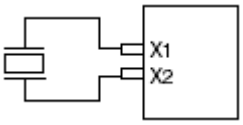
中断输出–$\overline{IRQ}$

这是一个中断信号输出端。该信号通知主机处理器，报警已经发生并请求动作。它是一个低电平有效的漏极开路输出端。

X1, X2

X1和X2脚分别用作反相放大器的输入和输出端。使用一个外部32.768kHz的石英晶体，为X1205提供一个实时时钟的基准时间。建议使用Citizen CFS206-32.768KDZF型的晶体。内部补偿电路被包含到一个完整的振荡电路中。注意晶体和电路的布线。强力推荐在器件周围布置大量的地线，X1和X2的连线应尽可能的短。更多细节看应用章节。

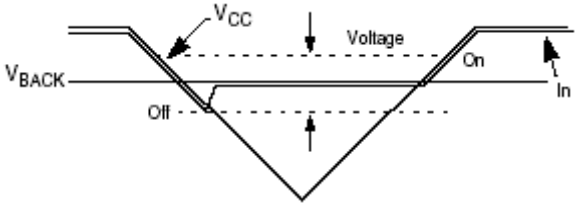
图1 推荐的晶体连接法



电源控制操作

电源控制电路认同一个V_{CC}和一个V_{BACK}输入，电源控制电路在V_{CC}< V_{BACK}– 0.2V时，采用V_{BACK}驱动时钟。当V_{CC}超过V_{BACK}时，切换回V_{CC}给器件供电。

图2 电源的控制



实时时钟工作

实时时钟（RTC）使用一个外部的 32.768kHz 石英晶体来保持精确的年、月、日、星期、时、分、秒。RTC 具有闰年校正功能。时钟对少于 31 日的月也能校正。有一位控制 24 小时或上午/下午格式。当 X1205 在 V_{CC} 和 V_{BACK} 都失掉以后再上电时，时钟即停止工作，直到在时钟寄存器中写入至少一个字节。

读实时时钟

通过一条读命令（Read）和在实时时钟的寄存器中规定相应的地址可以读 RTC。在序列读方式可以读 RTC 寄存器。因为时钟是连续运行的而一次读操作需要一定的时间，这就有可能在读操作过程中使时间改变。本器件中，由读命令将时间锁存在分离的锁存器中（时钟的下降沿在 RTC 数据输出之前的 ACK 位）。保持时钟连续运行，以避免当读操作时时间变化。当一次读出并不是由读操作引起时将发出报警。

写实时时钟

向 RTC 寄存器写入，可以设置时间和日期。为了避免当前时间被一次不完全的写操作改变，在 RTC 数据输入字节之前的 ACK 位时，时钟的下降沿将当前时间的值装载入分别的缓冲器中。时钟连续运行，新的串行输入数据取代缓冲器中的值。这个新的 RTC 值由有效写操作序列结束时的停止（stop）位装载回 RTC 寄存器。一次无效的写操作将中止时间

的更新，并且缓冲器的内容被丢弃。一次有效的写操作之后，RTC 将反映在停止（stop）位之后以第一个“一秒”时钟周期开始的最近装载的数据。RTC 寄存器写正在进行的同时 RTC 继续更新时间，并且在任何非易失性写操作时 RTC 继续运行。向 RTC 写入单个字节并不影响其它字节。

实时时钟的精度

实时时钟的精度依赖于石英晶体的频率，被用作RTC的基准时间。一个晶体引起共鸣的频率依赖于温度，RTC的性能也依赖于温度。晶体频率的偏差是来自晶体正常频率的温度反转函数。例如，大于20ppm频率偏差转变为一个每月大于1分钟的精度。这些参数来源于晶体生产厂家。Xicor公司的RTC系列可提供片内晶体补偿网络，当使用一个12.5 pF的负载晶体时，调整负载电容可改变振荡器频率，调整的范围是+116ppm 至–37 ppm。更多详细信息，请看应用章节。

时钟/控制寄存器（CCR）

时钟/控制寄存器位于一个区域，它只能跟随在从字节“1101111x”之后才能访问并且只能在地址[0000h:003Fh]读出或写入。时钟/控制存储器映象表包含从0000h到003Fh的存储器地址。定义地址在表1里有描述，未定义地址的写和读不推荐。

访问 CCR

CCR 的内容可以通过向 CCR 中的任何地址直接以字节写或页面写操作来修改。但是，在向 CCR（除了状态寄存器）写之前，WEL 和 RWEL 位，必须用一个两步的序列来设置好（见“向时钟/控制寄存器写”一节）。

CCR 被分成 5 段：它们是：

1．Alarm 0（8 字节，非易失性）
2．Alarm 1（8 字节，非易失性）
3．Control（4 字节，非易失性）
4．Real Time Clock（8 字节，易失性）
5．Status（1 字节，易失性）

每个寄存器通过缓冲器进行读和写。非易失性部分（或 RTC 的计数器部分）只有在 RWEL 被置位并且当一次有效的写操作和停止位之后才能更新。每次只能对 CCR 的一个段进行序列读或页面写操作。对另一段访问需要一次新的操作。连续的读或写一旦到达段的末端，将返回到段的开始再继续。读或写可以在 CCR 的任何地址开始。

不需要先设置 RWEL 位再写状态寄存器。第 5 段只支持单字节的读或写。连续读或写该段将停止操作。

任何时候通过完成一次在 CCR 中任何地址的随机读可以读出 CCR 的状态。这将返回那个寄存器地址的内容。其它的寄存器可通过完成一次连续读来读出。读指令将所有的时钟寄存器的内容锁存到一个缓冲器，因此时钟的更新不改变被读出的时间。对 CCR 的连续读不会导致从存储器阵列中输出数据。在读结束时，主机提供一个停止条件，以结束操作并释放总线。在读 CCR 之后，地址保留在先前的地址+1，因而用户可以执行 CCR 的当前地址读并继续读下一个寄存器。

表1　时钟/控制存储器映象图

Addr.	Type	Reg Name	Bit								Range	Default
			7	6	5	4	3	2	1	0		
003F	Status	SR	BAT	AL1	AL0	0	0	RWEL	WEL	RTCF		01h
0037	RTC (SRAM)	Y2K	0	0	Y2K21	Y2K20	Y2K13	0	0	Y2K10		20h
0036		DW	0	0	0	0	0	DY2	DY1	DY0	0-6	00h
0035		YR	Y23	Y22	Y21	Y20	Y13	Y12	Y11	Y10	0-99	00h
0034		MO	0	0	0	G20	G13	G12	G11	G10	1-12	00h
0033		DT	0	0	D21	D20	D13	D12	D11	D10	1-31	00h
0032		HR	MIL	0	H21	H20	H13	H12	H11	H10	0-23	00h
0031		MN	0	M22	M21	M20	M13	M12	M11	M10	0-59	00h
0030		SC	0	S22	S21	S20	S13	S12	S11	S10	0-59	00h
0013	Control (NONVOLATILE)	DTR	0	0	0	0	0	DTR2	DTR1	DTR0		00h
0012		ATR	0	0	ATR5	ATR4	ATR3	ATR2	ATR1	ATR0		00h
0011		INT	IM	AL1E	AL0E	0	0	X	X	X		00h
0010		0	0	0	0	0	0	0	0	0		00h
000F	Alarm1 (NONVOLATILE)	Y2K1	0	0	A1Y2K21	A1Y2K20	A1Y2K13	0	0	A1Y2K10		20h
000E		DWA1	EDW1	0	0	0	0	DY2	DY1	DY0	0-6	00h
000D		YRA1	Unused – Default = RTC Year value – Future expansion									
000C		MOA1	EMO1	0	0	A1G20	A1G13	A1G12	A1G11	A1G10	1-12	00h
000B		DTA1	EDT1	0	A1D21	A1D20	A1D13	A1D12	A1D11	A1D10	1-31	00h
000A		HRA1	EHR1	0	A1H21	A1H20	A1H13	A1H12	A1H11	A1H10	0-23	00h
0009		MNA1	EMN1	A1M22	A1M21	A1M20	A1M13	A1M12	A1M11	A1M10	0-59	00h
0008		SCA1	ESC1	A1S22	A1S21	A1S20	A1S13	A1S12	A1S11	A1S10	0-59	00h
0007	Alarm0 (NONVOLATILE)	Y2K0	0	0	A0Y2K21	A0Y2K20	A0Y2K13	0	0	A0Y2K10	19/20	20h
0006		DWA0	EDW0	0	0	0	0	DY2	DY1	DY0	0-6	00h
0005		YRA0	Unused – Default = RTC Year value – Future expansion									
0004		MOA0	EMO0	0	0	A0G20	A0G13	A0G12	A0G11	A0G10	1-12	00h
0003		DTA0	EDT0	0	A0D21	A0D20	A0D13	A0D12	A0D11	A0D10	1-31	00h
0002		HRA0	EHR0	0	A0H21	A0H20	A0H13	A0H12	A0H11	A0H10	0-23	00h
0001		MNA0	EMN0	A0M22	A0M21	A0M20	A0M13	A0M12	A0M11	A0M10	0-59	00h
0000		SCA0	ESC0	A0S22	A0S21	A0S20	A0S13	A0S12	A0S11	A0S10	0-59	00h

报警寄存器（Alarm0、Alarm1）

有两个报警寄存器，它们的内容由 RTC 寄存器的内容仿制，但增加了使能位并排除了 24 小时时间的选择位。使能位规定哪些寄存器可以用于报警寄存器和实时寄存器之间的比较。例如：

—结合其它使能位和一个专用报警时间，设置使能月位（MEOn*），用户能够在每一年里的相同时间设立一次触发报警。

*n=0 为报警 0；n=1 为报警 1

当发生一次匹配时，会设置一个报警标志。只有通过轮询 AL0 和 AL1 位或使能一个 $\overline{IRQ}$ 输出端来确定一次报警的发生。可将 $\overline{IRQ}$ 输出端用作一个硬件标志。

报警时使能位被放置在特殊功能寄存器的最高有效位。当所有使能位被设置为‘0’时，将没有报警。

—用户可以将 X1205 设置成每星期三的上午 8:00 报警。这可以将 EDWn、EHRn 和 EMNn 这三个使能位置“1”并将 DWAn、HRAn 和 MNAn 这三个报警寄存器设置为 8:00 AM Wednesday。

—每天下午 9:30 的报警由以下设置：EHRn 和 EMNn 使能位置“1”并将 HRAn 和 MNAn 寄存器设置为 9:30 PM。

*n=0 为报警 0；n=1 为报警 1

实时时钟寄存器（Real Time Clock）

时钟/日历寄存器（YR、MO、DT、HR、MN、SC）

这些寄存器采用 BCD 码表示时间。其中，SC（秒）、MN（分）的范围为 0 至 59，HR（时）在带有 AM 或 PM 显示（H21 位）时为 1 至 12，或者 0 至 23（当 MIL=1），DT（日期）为 1 至 31，MO（月）为 1 至 12，YR（年）为 0 至 99。

星期寄存器（DW）

该寄存器提供星期中的日期状态，它用三位 DY2 至 DY0 来表示星期中的 7 天。该计数值不断地作 0-1-2-3-4-5-6-0-1-2……的循环。数字值分配到星期中的某日是任意的，可以由系统软件设计者决定，缺省值定义为 0。

24 小时时间

如果 HR 寄存器中的 MIL 位为“1”，则 RTC 使用 24 小时格式。如果 MIL 位为“0”，则 RTC 使用 12 小时格式，这时位 H21 用作 AM/PM 指示，当 H21 为“1”显示 PM。时钟缺省为 H21=0 的标准时间。

闰年

闰年加一个 2 月 29 日，它是由年份数能被 4 除尽决定的。年份数能被 100 除尽不是闰年，除非它也能被 400 除尽。这表明 2000 年是闰年而 2100 年不是。X1205 没有校正 2100 年为闰年。

状态寄存器（SR）

状态寄存器位于 CCR 存储器映象图中，地址 003Fh。这是一个易失性寄存器，它用来控制写使能锁存位 WEL 和 RWEL，读两个电源状态和两个报警位。这个寄存器是独立于存储器阵列和时钟/控制寄存器（CCR）的。

表 2　状态寄存器（SR）

地址	7	6	5	4	3	2	1	0
003Fh	BAT	AL1	AL0	0	0	RWEL	WEL	RTCF
缺省	0	0	0	0	0	0	0	1

* BAT：电池供电——易失性

这位置“1”表示器件由电池 V_{BACK} 供电而不是由 V_{CC}。这是由硬件置位/复位（在 X1205 内部）的只读位。一旦器件开始由 V_{CC} 供电，则置该位为“0”。

* AL1、AL0：报警位——易失性

这两位显示报警 1 或报警 2 是否与实时时钟匹配。如果有某一个匹配了，则相应位被置“1”。在状态寄存器（SR）读操作中最后一个数据位的下降沿复位该标志。注意：当 SR 读时只有被置“1”的 AL 位会被复位。由一次报警发生的报警位置位时正当 SR 读操作发

生，则 AL 位将保持置位到读操作完成之后。

* RWEL：寄存器写使能锁存——易失性

该位是一易失性锁存位，上电时为“低”（禁止）状态。在向任何时钟/控制寄存器写入之前 RWEL 位必须先置“1”。写 RWEL 位并不引起一次非易失性写周期，因此在停止（stop）条件之后器件可以立刻准备下一次操作。向 CCR 写入需要 RWEL 和 WEL 这两位以一定的步骤都被置位。

* WEL：写使能锁存——易失性

WEL 位控制在一次写操作时对 CCR 和存储器阵列的访问。该位是易失性的，上电时处于“低”（禁止）状态。当 WEL 位为“低”时，对 CCR 或任何阵列地址写入都将被忽略（在数据字节之后将不发生应答）。通过对状态寄存器（SR）的 WEL 位写“1”对其它位写“0”来使 WEL 位置 1。一旦被置 1，WEL 保持置 1，直到被复位为 0（通过向 SR 的 WEL 位写“0”和其它位也写入“0”）或者直到器件再次上电。写 WEL 位并不引起一次非易失性写周期，因此在停止（stop）条件之后器件可以立刻准备下一次操作。

* RTCF：实时时钟失效位——易失性

在全部电源失效后该位被置“1”。这是一个由硬件置位（在 X1205 内部）的只读位，在器件失去全部电源后再上电时该位置位。该位的置位与先加 V_{CC} 或 V_{BACK} 无关。失去其中一个或另一个电源并不导致 RTCF 位的置位。向 RTC 的第一次有效写（只要写一个字节即可）即将 RTCF 位复位为“0”。

* 不用的位

位 3 和位 4 没有使用，但在这些位必须置“0”。当向 SR 读出数据字节时，将包含这二位的“0”。

中断控制寄存器（INT）

中断控制和状态位（IM、AL1E、AL0E）

有两个中断控制位：Alarm1 中断使能（AL1E）和 Alarm0 中断使能（AL0E）用来分别使能或禁止报警中断信号（$\overline{IRQ}$）的输出。当相应的位置“1”中断即被使能。

两个易失性位（AL1 和 AL0）各自与两个报警有关，用来指出某一报警是否已发生。这些位在报警条件时置位，而与报警中断是否使能无关。AL1 和 AL0 位在读状态寄存器时，由第 8 个时钟的下降沿复位为 0。

脉冲中断模式

脉冲中断模式容许重复或复发报警功能，因此一个重复或复发报警能够被设置到每N秒，或每N分钟，或每N小时，或每N日，或每个星期的同一天。脉冲中断模式被作为重复中断模式，重复速度由报警时间设定。

当IM位被设定时，脉冲中断模式被使能。一旦达到报警条件，报警 $\overline{IRQ}$ 输出脚将输出一个较短的持续时间(大约10～40ms)的单一脉冲。 假如中断模式位（IM位）被设置，那么这一脉冲将是周期性的。

IM位	中断 / 报警频率
0	报警单一时间事件的设定

1	报警重复 / 复发时间事件的设定

片内振荡器的补偿

数字微调寄存器（DTR）——DTR2、DTR1和DTR0（非易失性）

数字微调位 DTR2、DTR1 和 DTR0 调整每秒钟的计数值和平均 ppm 误差，以获取更好的精度。

DTR2 是一个符号位，当等于 1 时为正 ppm 补偿，而等于 0 时为负 ppm 补偿。

DTR1 和 DTR0 是刻度位，DTR1 给出的是 10ppm 调整，DTR0 给出的是 20ppm 调整。

通过这三位可以表示－30ppm 至+30ppm 的补偿范围。

表 3　数字微调寄存器

DTR 寄存器			预计 ppm 频率
DTR2	DTR1	DTR0	
0	0	0	0（默认）
0	1	0	+10
0	0	1	+20
0	1	1	+30
1	0	0	0
1	1	0	-10
1	0	1	-20
1	1	1	-30

模拟微调寄存器（ATR）——ATR5、ATR4…和 ATR0（非易失性）

从 ATR5 至 ATR0 六个模拟微调位用来调整片内负载电容，片内负载电容的范围从 3.25pf 至 18.75pf。每一位都有一个不同的电容调节比重，使用 Citizen CFS-206 晶体与不同的 ATR 位组合，可以对额定频率提供预计从+116ppm 至－37ppm 的 ppm 补偿范围。数字的和模拟的微调结合起来可以给出高达+146ppm 的调整。

片内电容可以通过下列公式计算出来：

C_{ATR} = [(ATR 值，十进制) x 0.25pF] + 11.0pF

注意ATR值以2的补码形式，ATR（000000）=11.0pF,因此整个范围是从3.25pF到18.75pF，每步调节0.25pF。

这些值是在典型值基础上计算出来的。由晶体控制的整个负载电容除ATR值以外还包括约为2pF的封装和电路板电容。

详情见应用部分说明和Xicor应用手册AN154。

向时钟/控制寄存器写

要改变时钟/控制寄存器的任何非易失性位，需要以下步骤：

——写 02h 至状态寄存器将“写使能锁存”（WEL）位置 1。这是一次易失性操作，所以在写后没有延迟。操作由“开始”（start）引导，由“停止”（stop）结束。

——写 06h 至状态寄存器，将“寄存器写使能锁存”（RWEL）和 WEL 这二位都置 1。这也是易失性操作。在数据字节中的 0 是必须的。操作由“开始”引导由“停止”结束。

——向时钟/控制寄存器写 1 至 8 个字节，写入所需的时钟、报警或控制数据。这个序列由“开始”位引导，需要一个从地址字节“11011110”和 CCR 中的一个地址，并由“停

止”位结束。向 CCR 写会改变非易失性寄存器的值，这会启动一次非易失性写，将耗时 10ms 才能完成。对没有定义的区域写入将不起作用。RWEL 位在一次非易失性写周期完成时将被复位，所以对 CCR 内容作另一次改变时，必须重复以上序列。如果由于任何原因而上述序列没有完成（例如，发送一个不正确的位数或发送“开始”而未发送“停止”等），则 RWEL 位未被复位，器件保持在激活方式。

——写全“0”至状态寄存器，将 WEL 和 RWEL 位都复位。

——在前面的任意操作间发生一次读操作，将不会中断寄存器的写操作。

串行通信

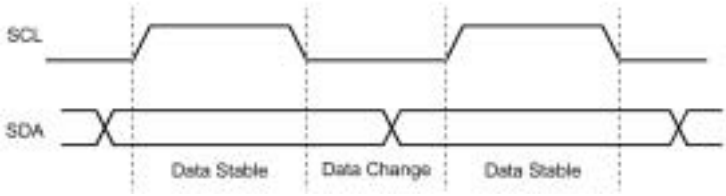
接口协议

器件支持双向总线协议。该协议定义向总线发送数据的任何器件为发送器，而接收数据的器件为接收器。控制发送的器件称为主机而被控制的器件称为从机。主机总是启动数据的传送，并提供用于发送和接收操作的时钟。所以，本系列器件在所有的应用中用作从器件。

时钟和数据

在 SDA 线上的数据只有当 SCL 为低（LOW）时才能改变状态。当 SCL 为高（HIGH）时改变 SDA 状态被用于表示“开始”（start）和“停止”（stop）条件，见图 3。

图 3　在 SDA 总线上有效的数据变化



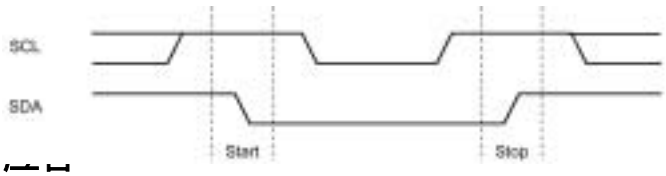
“开始”条件

所有的命令由“开始”条件引导，它是当 SCL 为高时 SDA 由高至低的变化。器件不断地监视 SDA 和 SCL 线上的开始条件，并且在该条件被满足以前不响应任何命令，见图 4。

“停止”条件

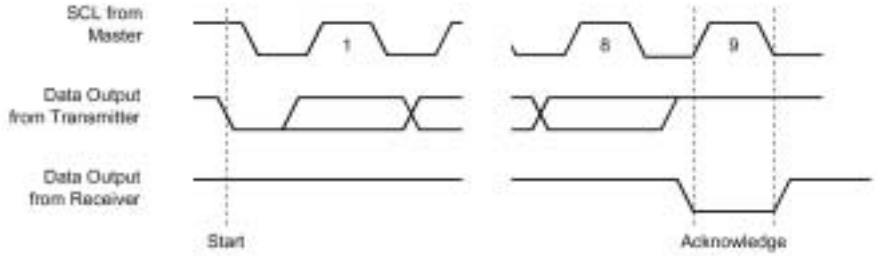
所有的通信必须由停止条件来终止，它是当 SCL 为高时 SDA 由低至高的变化。停止条件也用于在一次读序列之后将器件置于电源等待方式。在发送器件已经释放总线后才可以发出一个停止条件，见图 4。

图 4　有效的“开始”和“停止”条件



应答是一个用来表示数据传送成功的软件协议。发送器件（无论是主机或从机）在发送 8 位后将释放总线。在第九个时钟周期中，接收器将 SDA 线拉低作为对接收到 8 位数据的应答，请参见图 5。

图 5　接收器发出的应答



在器件辨识了开始条件以及如果在从地址字节中包含了正确的器件标识符和选择位后，将用一个应答来作为响应。如果选择一次写操作，在接收了每 8 位序列以后器件将以一个应答来作为响应。器件将会对所有收到的数据和地址字节作出应答，除非：

——从地址字节的器件辨识符和/或选择位是不正确的

——在写保护寄存器中的 WEL 位为低，而要写数据字节时

——状态寄存器写操作时有第二个数据字节（只允许一个数据字节）

在读方式中，器件将在发送 8 位数据后释放 SDA 线，然后监视该线上的应答信号。如果检测到一个应答并且主机没有产生“停止”条件，则器件将继续发送数据。如果没有检测到应答则器件将终止进一步的数据发送。主机这时必须立即发出一个停止条件，以便使器件返回等待方式并使器件处于一种已知的状态。

器件寻址

主机在发出开始条件后必须跟着输出一个从地址字节。从地址字节“1101”访问 CCR。

从地址字节的位 3 到位 1 确定器件的选择位，将其置位为“111”。

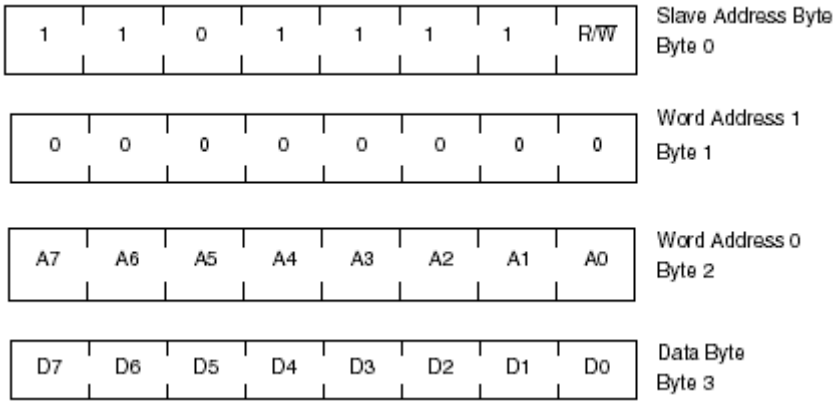
从地址字节的最后一位定义了要完成的操作。当 R/$\overline{W}$ 位为 1 时，选择了读操作；为 0 时选择写操作，见图 6。

从 SDA 总线输入了整个的从地址字节之后，X1205 将器件辨识符及器件选择位与“1101111”比较。如果比较正确，器件输出一个应答至 SDA 线。

在从地址字节之后跟随一个两字节的字地址。这个字地址可以由主机提供也可以从内部计数器获得。在上电时内部地址计数器被设置为地址 0h，因此 CCR 阵列的当前地址读从地址 0 开始。当需要时，如同随机读那样，主机必须如图 6 所示提供两字节的字地址。

在随机读操作时，“伪”写操作中的从地址字节，必须与“读操作”时的从地址字节一致。即如果对时钟/控制寄存器（CCR）的随机读，则在两种情况下的从地址字节必须为 1101111x。

图 6　从地址、字地址和数据字节

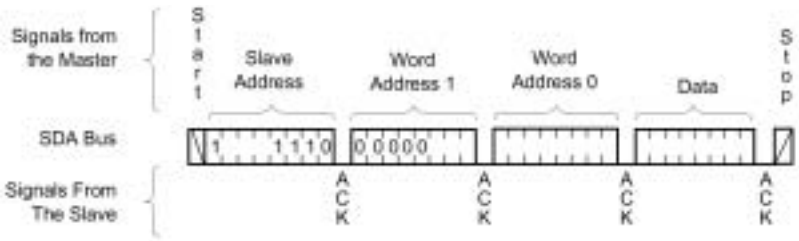


写操作

字节写

要进行一次写操作，器件需要从地址字节和字地址字节。这使得主机可以访问 CCR 阵列中的任何一个字。（注意：在写 CCR 之前，主机必须先向状态寄存器写 02h，然后 06h 以使能写操作。见“向时钟/控制寄存器写”节中的说明。）收到每个地址字节后，X1205 响应一个应答。在两个地址字节都收到之后，X1205 等待 8 位数据。在收到 8 位数据之后，X1205 再响应一个应答。然后主机通过产生一个停止条件来终止传送。X1205 接着开始一次将数据写入非易失性存储器的内部写周期。在内部写周期时，器件的输入是被禁止的，所以器件将不响应来自主机的任何请求。SDA 输出端这时处于高阻抗状态，见图 7。

图 7　字节写序列



停止和写方式

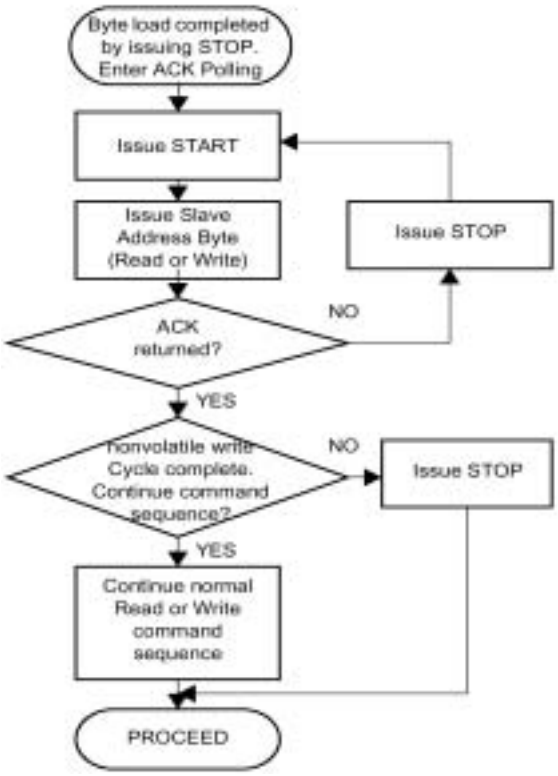
主机至少在送出一个完全的数据字节和收到相关的 ACK 信号以后，才可以送出停止来终止写操作。如果在数据字节的中间或在一个完全的字节和 ACK 之前发生一个停止条件，则 X1205 将自动复位而不完成写操作。阵列的内容不受影响。

应答查询

在非易失性写周期时禁止输入通常要用 5ms 写周期时间。一旦发出停止条件表示主机字节写操作结束时，X1205 即开始内部非易失性写周期。应答查询即可立即开始。为了做到这点，主机发出一个开始条件跟随一个用来写或读操作的从地址字节。如果 X1205 仍然

忙于非易失性写周期，则将不返回应答（ACK）。当 X1205 已经完成写操作，将返回一个 ACK，而主机即可进行读或写操作。请参见图 9 中的流程图。

图 9　应答查询序列



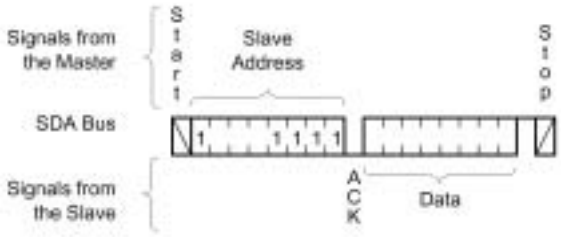
读操作

有三种基本的读操作：当前地址读、随机读和连续读。

当前地址读

X1205 内部有一个地址计数器，它保持的地址是最后一次读出的字节的地址加一。所以如果最后一次读的地址是 n，下一次读操作将访问地地址 n+1。在收到将 R/$\overline{W}$ 位设置为 1 的从地址字节后，X1205 发出一个应答，然后发送 8 位数据。主机在第九个时钟时不响应一个应答而是发出一个停止条件来终止读操作。见图 8 中关于地址，应答和数据传送序列。

图 8 当前地址读序列



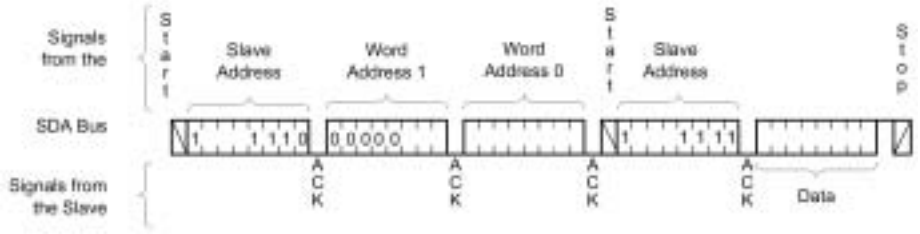
必须注意：读操作的第九个时钟周期并不是“不必关心”的。为了终止读操作，主机必须在第九个周期发出停止条件或者将 SDA 线拉高，然后再发停止条件。

随机读

随机读操作允许主机访问 X1205 中的任何地址。在发出将 R/$\overline{W}$ 位置 0 的从地址字节之前，主机必须首先完成一次“伪”写操作。

主机发出开始条件和从地址字节，收到一个应答，然后发出字地址字节。在收到每个字地址字节的应答后，主机立即发出另一个开始条件和将 R/$\overline{W}$ 位置 1 的从地址字节。之后器件发出一个应答接着就是 8 位数据。主机要终止读操作时就不响应一个应答而发一个停止条件。关于地址、应答和数据传送序列请参见图 10。

图 10 随机地址读序列



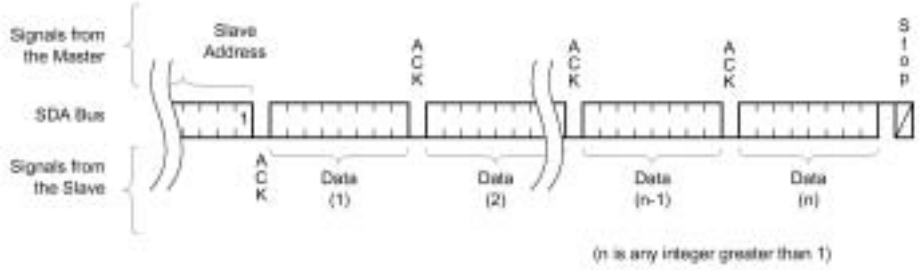
有一种类似的操作称为“设置当前地址”，器件设置地址后在图 10 中第二个开始条件处主机再发一个停止条件。停止条件后 X1205 即进入等待方式，并且直到在检测到一个开始条件前对总线上所有的活动都不理睬。这个操作在地址计数器中加载了新的地址。下一次当前地址读操作将从新的加载地址读出。这种读操作对主机知道它所需读出的下一个地址但数据尚未准备好的情况是有用的。

连续读

连续读可以由当前地址读或是随机地址读任一种方法来启动。第一个数据字节可以用其中一种方式发送；但是现在主机响应一次应答，以表示它需要随后的数据。器件对每一个接收到的应答继续输出数据。主机若要终止读操作，则不响应应答而发出一个停止条件。

数据输出是顺序的，如从地址 n 输出数据后面跟着从地址 n+1 输出数据。用于读操作的地址计数器逐次加 1，通过所有的页和行地址，可以在一次操作中将全部存储器的内容系列地读出。在地址空间的终端，计数器“返回”到地址空间的开始端，X1205 继续在收到每个应答时输出数据。关于应答和数据传送序列请参见图 11。

图 11 系列读序列



极限参数

工作温度	−65°C 至+135°C
储存温度	−65°C 至+150°C

V_{CC}、V_{BACK}和 $\overline{IRQ}$ 管脚上的电压（相对于地）	－0.5V 至 7.0V
SCL、SDA、X1 和 X2 管脚上的电压（相对于地）	－0.5V 至 7.0V 或 0.5V
V_{CC} 或 V_{BACK}（两者中的较高者）上的直流输出电流	5mA
引线温度（焊接，10 秒）	300°C

注：强度超出所列的极限参数可能导致器件的永久性损坏。这些仅仅是极限参数，并不意味着在极限条件下或在任何其它超出推荐工作条件所示参数的情况下器件能有效地工作。延长在极限参数条件下的工作时间会影响器件的可靠性。

直流工作特性（温度=－40°C 至+85°C，除非另有说明。）

Symbol	Parameter	Conditions	Min	Typ	Max	Unit	Notes
V_{CC}	Main Power Supply		2.7		5.5	V	
V_{BACK}	Backup Power Supply		1.8		5.5	V	
V_{CB}	Switch to Backup Supply		V_{BACK} -0.2		V_{BACK} -0.1	V	
V_{BC}	Switch to Main Supply		V_{BACK}		V_{BACK} +0.2	V	

工作特性

Symbol	Parameter	Conditions	Min	Typ	Max	Unit	Notes
I_{CC1}	Read Active Supply Current	V_{CC} = 2.7V			400	µA	1, 5, 7, 14
		V_{CC} = 5.0V			800	µA	
I_{CC2}	Program Supply Current (nonvolatile)	V_{CC} = 2.7V			2.5	mA	2, 5, 7, 14
		V_{CC} = 5.0V			3.0	mA	
I_{CC3}	Main Timekeeping Current	V_{CC} = 2.7V			10	µA	3, 7, 8, 14, 15
		V_{CC} = 5.0V			20	µA	
I_{BACK}	Timekeeping Current	V_{BACK} = 1.8V		1.25		µA	3, 6, 9, 14, 15 "See Performance Data"
		V_{BACK} = 3.3V		1.5		µA	
I_{LI}	Input Leakage Current				10	µA	10
I_{LO}	Output Leakage Current				10	µA	10
V_{IL}	Input LOW Voltage		-0.5		V_{CC} x 0.2 or V_{BACK} x 0.2	V	13
V_{IH}	Input HIGH Voltage		V_{CC} x 0.7 or V_{BACK} x 0.7		V_{CC} + 0.5 or V_{BACK} + 0.5	V	13
V_{HYS}	Schmitt Trigger Input Hysteresis	V_{CC} related level	.05 x V_{CC} or .05 x V_{BACK}			V	13
V_{OL}	Output LOW Voltage for SDA/IRQ	V_{CC} = 2.7V			0.4	V	11
		V_{CC} = 5.5V			0.4		

注：（1）在“开始”条件后器件进入激活状态并保持激活：9 个时钟周期，如果在从地址字节中的器件选择位不正确或在读或写操作的“停止”条件后 200ns

（2）在写操作的“停止”条件后 200ns 器件进入编程状态并要持续 t_{wc}

（3）在任何“停止”条件后器件进入计时状态 200ns，除非启动一次非易失性写周期；“停止”条件后启动一次非易失性写周期 t_{wc}；在“开始”条件后若在从地址字节中未跟随正确的器件选择位则计时 9 个时钟周期

（4）仅供参考，未测试

（5）$V_{IL}=V_{CC}\times 0.1$，$V_{IH}=V_{CC}\times 0.9$，f_{SCL}=400kHz

（6）V_{CC}=0V

（7）V_{BACK}=0V

www.icbase.com

（8）V_{SDA}=V_{SCL}=V_{CC}，其它=GND 或 V_{CC}
（9）V_{SDA}=V_{SCL}=V_{BACK}，其它=GND 或 V_{BACK}
（10）V_{SDA}=GND 至 V_{CC}，V_{CLK}=GND 或 V_{CC}
（11）I_{OL}=3.0mA（5V 时），1.5mA（2.7V 时）
（12）I_{OH}=－1.0mA（5V 时），－0.4mA（2.7V 时）
（13）门限电压基于较高的 V_{CC} 或 V_{BACK}
（14）在 X1、X2（25°C）处用推荐的晶体和振动器网络
（15）工作在环境温度 25°C 下的典型值

电容（T_A=25°C，f=1.0MHz，V_{CC}=5V）

Symbol	Parameter	Max.	Units	Test Conditions
C_{OUT}(1)	Output Capacitance (SDA, $\overline{IRQ}$)	10	pF	V_{OUT} = 0V
C_{IN}(1)	Input Capacitance (SCL)	10	pF	V_{IN} = 0V

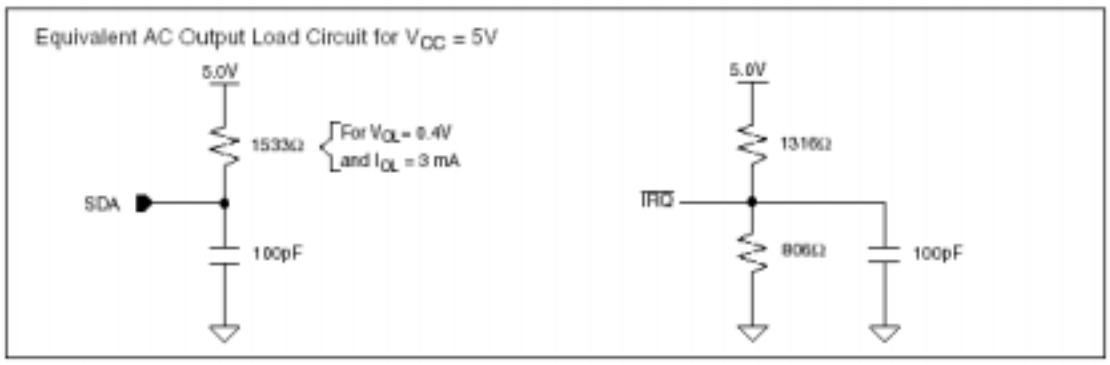
注：　（1）该参数周期性抽样，并非 100%测试
（2）X1 和 X2 引脚间的输入电容可用模拟微调寄存器在 5pF 至 19.75pF 范围内变动。

交流特性

交流测试条件

Input pulse levels	V_{CC} x 0.1 to V_{CC} x 0.9
Input rise and fall times	10ns
Input and output timing levels	V_{CC} x 0.5
Output load	Standard output load

图 14. V_{CC}=5.0V，测试器件时的标准输出负载



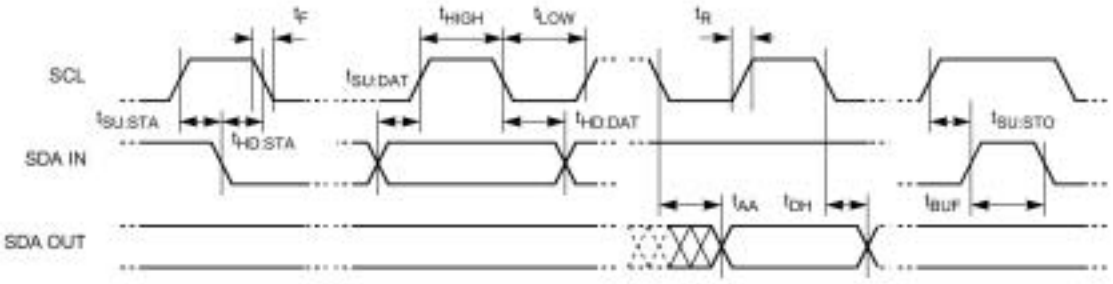
交流特性参数表（T_A=－40°C 至+85°C，V_{CC}=+2.7V 至+5.5V，除非另有规定。）

Symbol	Parameter	Min.	Max.	Units
f_{SCL}	SCL Clock Frequency		400	kHz
t_{IN}	Pulse width Suppression Time at inputs	50(1)		ns
t_{AA}	SCL LOW to SDA Data Out Valid		0.9	µs
t_{BUF}	Time the bus must be free before a new transmission can start	1.3		µs
t_{LOW}	Clock LOW Time	1.3		µs
t_{HIGH}	Clock HIGH Time	0.6		µs
$t_{SU:STA}$	Start Condition Setup Time	0.6		µs
$t_{HD:STA}$	Start Condition Hold Time	0.6		µs
$t_{SU:DAT}$	Data In Setup Time	100		ns
$t_{HD:DAT}$	Data In Hold Time	0		µs
$t_{SU:STO}$	Stop Condition Setup Time	0.6		µs
t_{DH}	Data Output Hold Time	50		ns
t_R	SDA and SCL Rise Time	20 +.1Cb(1)(2)	300	ns
t_F	SDA and SCL Fall Time	20 +.1Cb(1)(2)	300	ns
Cb	Capacitive load for each bus line		400	pF

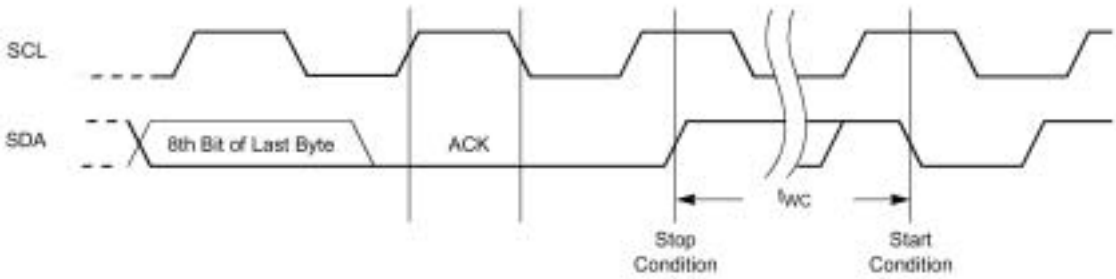
注：（1）该参数定期抽样而不是 100%测试
（2）Cb=一根总线的总电容，单位 pF

时序图

总线时序



写周期时序



上电时序

Symbol	Parameter	Min.	Typ.(2)	Max.	Unit
t_{PUR}(1)	Time from Power Up to Read			1	ms
t_{PUW}(1)	Time from Power Up to Write			5	ms

注：（1）由 V_{CC} 稳定开始直到规定的操作能够启动这段时间。这些参数定期抽样并非 100% 测试 V_{CC} 的回转速度在 0.2mV/μs 至 50mV/μs 之间

（2）典型值是在 T_A=25°C 和 V_{CC}=5.0V 时的值

非易失性写周期时序

Symbol	Parameter	Min.	Typ.(1)	Max.	Unit
t_{WC}[1]	Write Cycle Time		5	10	ms

注：（1）t_{wc} 是从有效的写序列结束时的停止条件至内部非易失性写周期结束之间所需的时间。对用户来说这是任何非易失性写允许的最小时间，除非使用应答查询

应用部分

晶体振荡器和温度补偿

Xicor公司在芯片上集成了振荡器补偿电路，消除了对外部元件需求和过温引起晶体漂移的调节，具有高精度的时间保持（<5ppm温漂）。

Xicor公司实时时钟系列使用同一个带片内晶体补偿网络的振荡器电路，内含可调负载电容。晶体是唯一需要的外部元件。最优化的补偿网络可在带有某些晶体参数的应用中使用，这些参数在现有的许多表面贴或直插的晶体中通用。表6总结了这些参数。

表7.包含了一些晶体制造厂家和型号，可以满足Xicor公司实时时钟器件的要求。

在表6里描述的反向温度是指随温度曲线变化的漂移达到最高峰值时的温度。这一曲线相对于$(T-T0)^2$增量的漂移。例如，Epson公司的器件MC-405，反向温度的典型值为25℃，在温度极限－40℃和85℃上，会产生>110ppm的尖峰漂移。调节晶体负载电容使改变偏移成为可能，使对晶体频率的改变可控。

Xicor公司实时时钟系列通过器件上负载电容的调整，以达到在温度上的调节，这一控制来自于模拟调节寄存器或ATR，它有6个控制位。ATR电路的负载电容范围大约为3.25pF～18.75pF，每次增量为0.25 pF。

注意实际电容量也应该包括与封装相关的2pF电容。采用商用晶体进行的在线路测试表明该范围内的电容允许控制+116ppm ～–37ppm范围内的频率，采用一个12.5pF的负载晶体。

除了由可调负载电容提供的模拟补偿以外，数字补偿的这一特性也适用于Xicor公司实时时钟的所有系列。数字微调寄存器简称DTR，有三位组成。它的操作是在时钟信号里增加或减小脉冲，范围是±30ppm，每档的增量为10ppm，默认值为0ppm。DTR控制器被用于在温度上频率漂移的粗调和晶体初始精度的修正。

表6　Xicor实时时钟晶体参数的需求

Parameter	Min	Typ	Max	Units	Notes
Frequency		32.768		kHz	
Freq. Tolerance			±100	ppm	Down to 20ppm if desired
Turnover Temperature	20	25	30	°C	Typically the value used for most crystals
Operating Temperature Range	-40		85	°C	
Parallel Load Capacitance		12.5		pF	
Equivalent Series Resistance			50	kΩ	For best oscillator performance

表7　晶体制造厂家

Manufacturer	Part Number	Temp Range	+25°C Freq Toler.
Citizen	CM201, CM202, CM200S	-40 to +85°C	±20ppm
Epson	MC-405, MC-406	-40 to +85°C	±20ppm
Raltron	RSM-200S-A or B	-40 to +85°C	±20ppm
SaRonix	32S12A or B	-40 to +85°C	±20ppm
Ecliptek	ECPSM29T-32.768K	-10 to +60°C	±20ppm
ECS	ECX-306/ECX-306I	-10 to +60°C	±20ppm
Fox	FSM-327	-40 to +85°C	±20ppm

ATR(模拟微调寄存器)控制器连同温度传感器芯片是应用在高精度的电路校准，同备用电池一起上电的实时时钟，PHZ输出被设置为32.768kHz，测量频率的漂移。调整ATR控制器的设置，使频率的漂移量最小。一次调整到一个精确的温度后，对其它离散温度，为使温漂降到最低，也可以进行调整。结果存入EEPROM。用这种方法，可以使温漂降到最低。Xicor公司评估板包含这一电路，可以实现这一功能。

更多详细的操作，可上Xicor公司网站www.xicor.com，参考应用笔记AN154。

印刷电路板布局的考虑

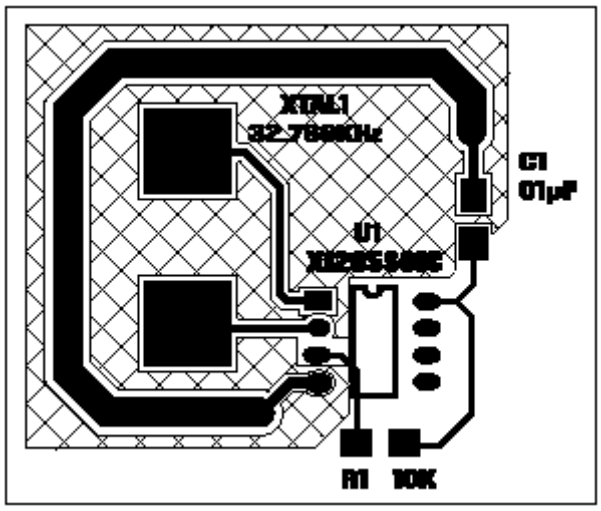
晶体输入X1有非常高的阻抗，它将从其它电路获得非常高的频率信号，晶体的另外一边连接到X2管脚上，它也是一个敏感点。这一信号能够耦合到振荡器电路中，引起双倍的时钟或错误的时钟，严重的影响实时时钟的精度。在实时时钟的电路板布局上，应关注其设计，以避免噪声的引入，图15表示X1205 SOIC布局的建议。

X1和X2连接到晶体的线应尽可能的短，考虑到减少噪声的侵蚀，围绕着晶体有一根很宽的地线，但在X1和X2附近不应接地线，因为它会增加管脚的负载电容。在实时时钟器件附近的PCB板的布局上应遵循这一指导方针。在芯片的V_{CC}管脚上添加一个非常小的退耦电容。电容的另一端接地。

装配

许多电子电路没有涉及到装配的问题，实时时钟器件的装配包括直插或热焊接一块充电电池到无源电路上。假如插座已经焊接到电路板上，电池在最后的总装时被插入。则对实时时钟的操作不会有任何问题。假如电池被直接焊接到电路板上，于是，实时时钟器件Vback管脚，由于焊接工具或间断的电池连接的原因，将会使振荡电路停振。解决的方法是电池一次性焊接到电路板上以后，把Vback管脚瞬间接地，将会使振荡电路起振。

图 15　Xicor 公司 SO-8 封装实时时钟布线图



振荡器的测量

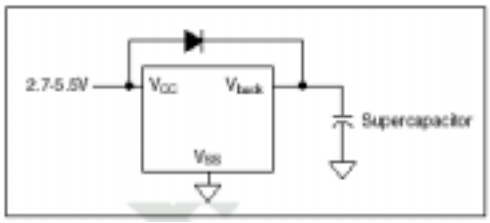
当选择一个正确的晶体并参照以上布线指南，大多数电路的振荡器起振小于1秒钟。有一些电路稍微长些，但起振明确产生的时间小于5秒种。当检测实时时钟电路时，普遍采用示波器探头接触到X2管脚（振荡器输出）上，观察其波形。请不要这样测试！尽管采用此方法，你可以看到一个有效的波形，应归于寄生效应（通常对地电容为10pF）。与实际的振荡电路相比，这一波形将是没有使用价值的信息，X2输出对容抗是非常灵敏的。因而电平和频率将受示波器探头寄生电容的影响。使用示波器探头可能会启动有故障的振荡器并会隐藏其它问题（尽管Xicor的实时时钟内部电路在使用适当的晶体和布线时可确保启动）。

分析实时时钟的最好方法是上电以后，读取实时时钟，看时间有无变化；或者假如芯片有PHZ输出，用示波镜检测PHZ管脚的输出（用控制寄存器使能它以后。由于该管脚为漏极开路，因此检测时需添加上拉电阻。）另外，X1226/1286/1205器件有一个IRQ输出，该输出可通过设置一个每分钟的报警来检测。如果用脉冲中断模式设置该报警，则每分钟一次的中断功能可用于确定振荡是否正确。

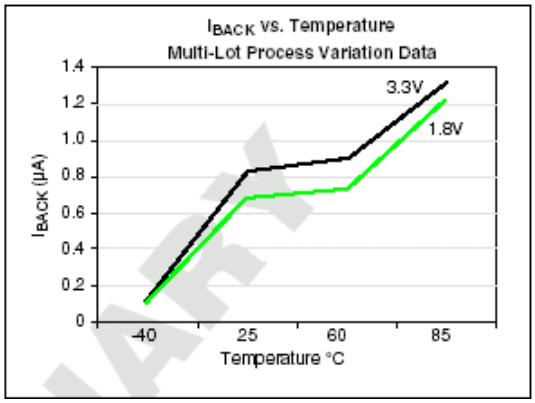
后备电池的操作

许多电池的类型都能够用于Xicor公司实时时钟产品，3.0V或3.6V的锂离子电池较为适合，使用期限为10年。另外的用法是在V_{CC}可能间歇性地丢失的应用中，可选择使用一个大容量的电容，备份时间可持续几天至两个星期，时间的长短依赖与电容的容量(>1F)大小。用一个简单的硅或肖特基二极管连接到Vcc以对大容量电容充电，该电容连接到Vback管脚，不能使用二极管对电池充电（特别是锂离子电池）。

图16　使用大容量电容的充电电路



I_{BACK}的性能数据



因为切换到电池的条件是 Vcc=Vback-0.1V(参见图16)，在正常操作期间，电池电压必须低于Vcc电压，否则电池将被逐步耗尽。在表8里给出了备用电池操作的条件：

表8 备份电池的操作

1. 应用实例, Vcc=5V, Vback=3.0V

Condition	Vcc	Vback	Vtrip	Iback	Notes
a. Normal Operation	5.00	3.00	4.38	<<1µA	
b. Vcc on with no battery	5.00	0	4.38	0	
c. Backup Mode	0–1.8	1.8-3.0	4.38	<2µA	Timekeeping only

2. 应用实例, Vcc=3.3V, Vback=3.0V

Condition	Vcc	Vback	Vtrip	Iback	
a. Normal Operation	3.30	3.00	2.65	<<1µA	
b. Vcc on with no battery	3.30	0	2.65	0	
c. Backup Mode	0–1.8	1.8–3.0*	2.65	<2µA*	Timekeeping only
d. UNWANTED - Vcc ON, Vback powering	2.65 - 3.30	> Vcc	2.65	up to 3mA	Internal Vcc=Vback

*Vback大于2.65V，高于Vtrip，电池将给整个器件供电

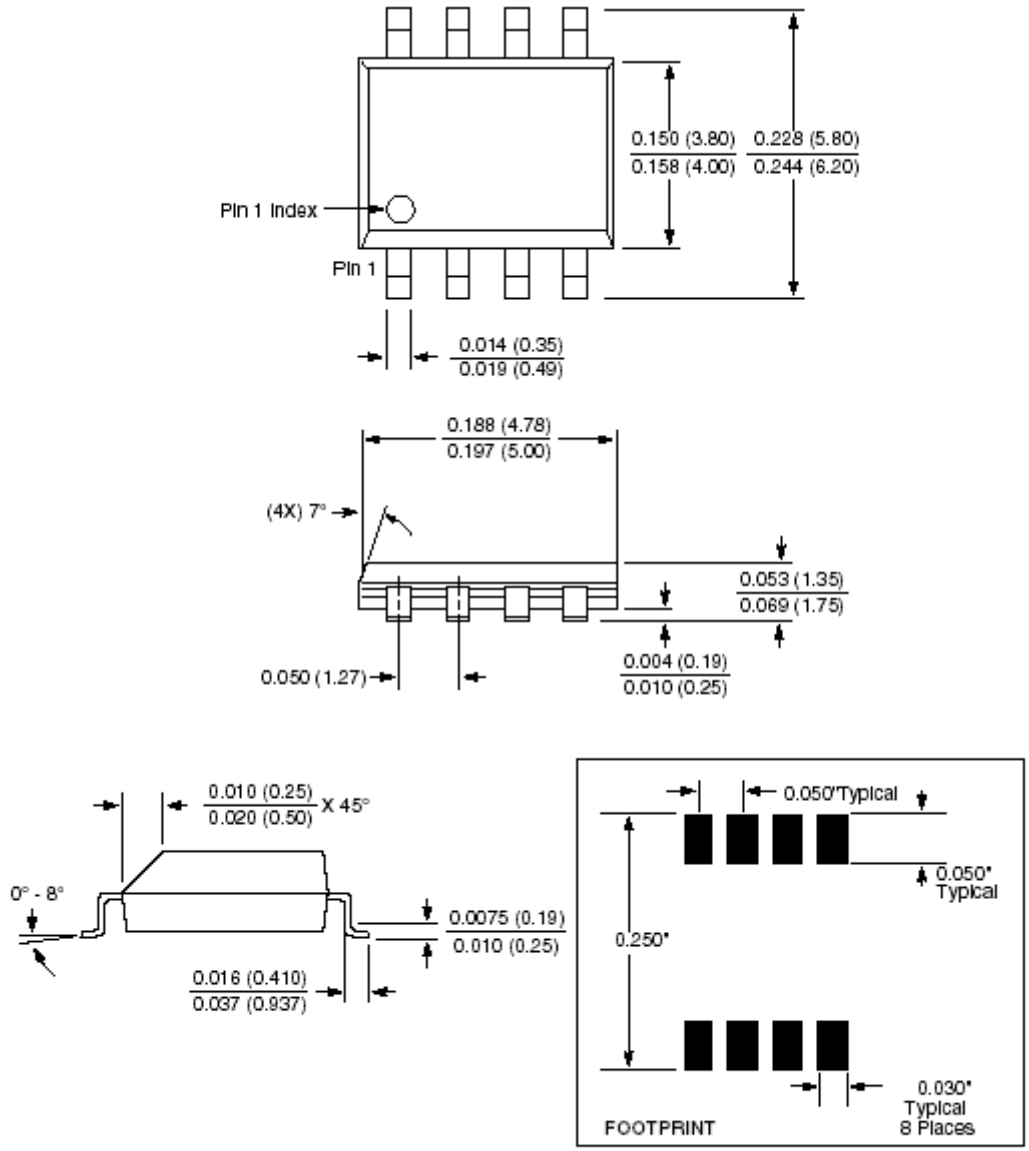
参照图16，加Vtrip到“内部Vcc”的节点驱动整个器件，这意味着假如Vcc掉电，并且Vback的电池电压高于Vtrip电压，然后整个芯片运行在电池供电的状态。假如Vback的电池电压低于Vtrip电压，然后整个芯片关闭，除振荡器和时间保持电路以外，所有输出功能将丧失。不必担心芯片可由Vback 供电，因为在这种模式下实时时钟器件的待机电流<2µA（在该数据手册中称作“主要的时间保持电路”） 。只有在串行接口被激活时，电源电流才会增加，在VCC掉电时，串行接口很可能进入停止状态。

为防止备用电池电压高于Vtrip电压，可加一个二极管压降（最好是硅极管）到电池上以确保电池电压低于Vtrip电压。这也可提供需要安全代理商批准的反向漏电流保护功能。

应避免使实时时钟器件在Vback同时大于Vcc和Vtrip时工作（表8中的第二种情形）。这会使电池很快被耗尽，因为在串行总线通信和非易失性写操作时将需要更大的电源电流。

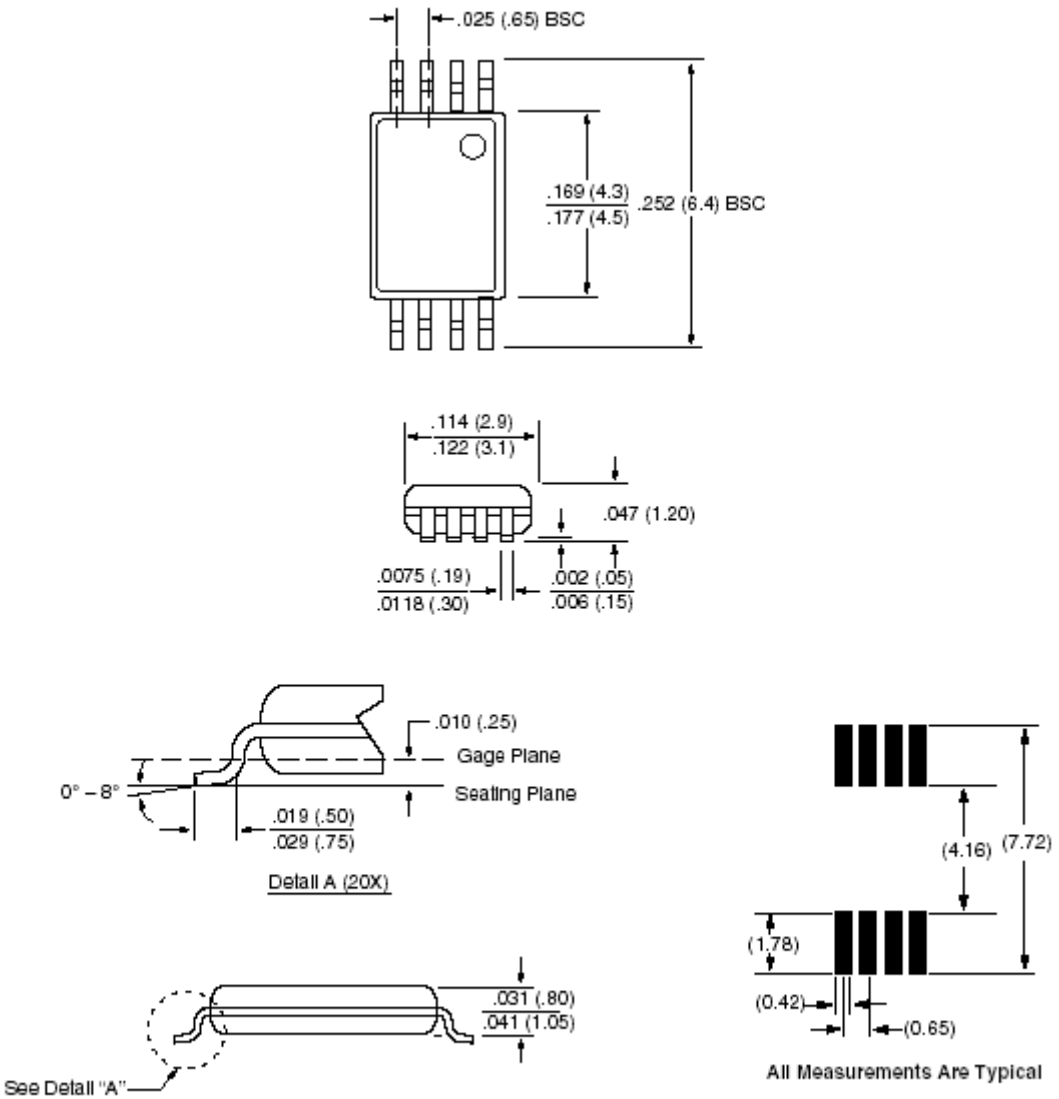
封装信息

8引脚塑料，SOIC，封装编码S8



NOTE: ALL DIMENSIONS IN INCHES (IN PARENTHESES IN MILLIMETERS)

8引脚塑料，TSSOP，封装编码S8



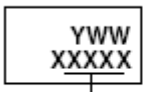
NOTE: ALL DIMENSIONS IN INCHES (IN PARENTHESES IN MILLIMETERS)

订购信息

V_{CC} Range	Package	Operating Temperature Range	Part Number
2.7-5.5V	8L SOIC	0–70°C	X1205S8
		-40–85°C	X1205S8I
	8L TSSOP	0–70°C	X1205V8
		-40–85°C	X1205V8I

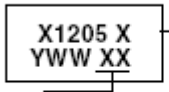
器件标识信息

8-Lead TSSOP



1205 = 2.7 to 5.5V, 0 to +70°C
1205I = 2.7 to 5.5V, -40 to 85°C

8-Lead SOIC



Blank = 2.7 to 5.5V, 0 to +70°C
I = 2.7 to 5.5V, -40 to 85°C

MCS-51系列单片机与X1205接口程序范例:（本示例仅供设计参考，力源公司不为该示例程序承担各种商业和法律责任）

```
;*************************************************************************/
;*    Copyright Wuhan P&S Info-Service Co., Ltd.                          */
;*************************************************************************/
;*                                                                        */
;*    This    program is the sample of X1205.                             */
;*                                                                        */
;*    Time : 2003.11.1                                                    */
;*    MCU    : AT89C52                                                    */
;*    Fosc : 11.0592M                                                    */
;*                                                                        */
;*************************************************************************/

;*********************************************
;内部数据RAM初始化定义?                        *
;*********************************************

;1)内部位寻址区RAM数据定义00H～7FH（对应字节为20H～2FH）
;I2C  标志
ACK       BIT  00H


;2)内部直接寻址区RAM数据定义30H～7FH
```

```
;   其中30H～3F  对应堆栈指针  30H<=SP<=3FH
CCR_DDR     EQU       40H ;时间存储首地址"40H"
TIME0    EQU       70H ;通用寄存器
TIME1    EQU       71H ;通用寄存器


;3)MCU 89C52'S I/O PORT DEFINED
SCL       EQU       P1.6
SDA       EQU       P1.7

;-----------------------------------------------
;(1) I2C PROGRAM                          |
;-----------------------------------------------
;a) COMMON PART

;启动I2C总线子程序

START:
      SETB     SDA
      NOP
      SETB     SCL
      NOP
      NOP
      NOP
      NOP
      NOP
      CLR      SDA
      NOP
      NOP
      NOP
      NOP
      NOP
      CLR      SCL
      NOP
      RET

;结束I2C总线子程序

STOP:
      CLR      SDA
      NOP
      SETB     SCL
      NOP
      NOP
```

```
    NOP
    NOP
    NOP
    SETB    SDA
    NOP
    NOP
    NOP
    NOP
    RET

;发送应答信号

MACK:
    CLR     SDA
    NOP
    NOP
    SETB    SCL
    NOP
    NOP
    NOP
    NOP
    NOP
    CLR     SCL
    NOP
    NOP
    RET

;发送非应答信号
MNACK:
    SETB    SDA
    NOP
    NOP
    SETB    SCL
    NOP
    NOP
    NOP
    NOP
    NOP
    CLR     SCL
    NOP
    NOP
    RET

;检验应答子程序
```

```
;返回值,ACK=1时表示有应答

CACK:
    SETB    SDA
    NOP
    NOP
    SETB    SCL
    CLR     ACK
    NOP
    NOP
    MOV     C,SDA
    JC      CEND
    SETB    ACK
CEND:
    NOP
    CLR     SCL
    NOP
    RET

;发送字节子程序
;字节数据放入ACC
;每发送一字节要调用一次CACK子程序,取应答位

WRBYTE:
    MOV     R0，#08H
WLP:
    RLC     A       ;取数据位
    JC      WR1
    SJMP    WR0     ;判断数据位
WLP1:
    DJNZ    R0，WLP
    NOP
    RET
WR1:
    SETB    SDA;    发送“1”
    NOP
    SETB    SCL
    NOP
    NOP
    NOP
    NOP
    NOP
    CLR     SCL
    SJMP    WLP1
```

```
WR0:
    CLR     SDA     ;发送“0”
    NOP
    SETB    SCL
    NOP
    NOP
    NOP
    NOP
    NOP
    NOP
    CLR     SCL
    SJMP    WLP1

;读取字节子程序
;读出的值在ACC
;每取一个字节要发送一个应答 / 非应答信号

RDBYTE:
    MOV     R0，#08H
RLP:
    SETB    SDA
    NOP
    SETB    SCL     ;时钟线为高，接收数据
    NOP
    NOP
    MOV     C,  SDA     ;读数据位
    MOV     A,  R2
    CLR     SCL         ;将SCL拉低，时间大于4.7us
    RLCA                ;进行数据位处理
    MOV     R2，A
    NOP
    NOP
    NOP
    DJNZ    R0，RLP     ;未够8位，再来一次
    RET
```

;b)X1205的读写部分子程序

```
;X1205的写实时时钟程序(依次设置秒/分/时/日/月/年/周/千年)
;秒/分/时/日/月/年/周/千年==>以CCR_DDR为首地址的连续8字节中
;占用：A、R3、CY、R1

WRITE_CCR:
    LCALL   START                   ;启动总线
    MOV     A,  #0DEH               ;X1205器件地址
    LCALL   WRBYTE                  ;发送器件从地址
    LCALL   CACK
    JNB     ACK,    RETWRNCCR       ;无应答则退出
    MOV     A,  #00H                ;指定子地址高字节
    LCALL   WRBYTE
    LCALL   CACK
    MOV     A,  #3FH                ;指定子地址低字节（状态寄存器地址）
    LCALL   WRBYTE
    LCALL   CACK
    MOV     A,  #02H                ;发02H到状态寄存器，WEL位置“1”
    LCALL   WRBYTE
    LCALL   CACK
    LCALL   STOP

    LCALL   START                   ;启动总线
    MOV     A,  #0DEH               ;X1205器件地址
    LCALL   WRBYTE                  ;发送器件从地址
    LCALL   CACK
    JNB     ACK,    RETWRNCCR       ;无应答则退出
    MOV     A,  #00H                ;指定子地址高字节
    LCALL   WRBYTE
    LCALL   CACK
    MOV     A,  #3FH                ;指定子地址低字节（状态寄存器地址）
    LCALL   WRBYTE
    LCALL   CACK
    MOV     A,  #06H                ;发06H到状态寄存器，RWEL和WEL位置“1”
    LCALL   WRBYTE
    LCALL   CACK
    LCALL   STOP

    MOV     R3, #08H                ;写入时间初始值“秒/分/时/日/月/年/周/千年”
    LCALL   START                   ;启动总线
    MOV     A,  #0DEH               ;X1205器件地址
    LCALL   WRBYTE                  ;发送器件从地址
    LCALL   CACK
    JNB     ACK,    RETWRNCCR           ;无应答则退出
```

```
    MOV     A,   #00H                   ;指定子地址高字节
    LCALL   WRBYTE
    LCALL   CACK
    MOV     A,   #30H                   ;指定子地址低字节(实时时钟地址）
    LCALL   WRBYTE
    LCALL   CACK
    MOV     R1,  #CCR_DDR
WRCCRDA:
    MOV     A,   @R1
    LCALL   WRBYTE                      ;开始写数据
    LCALL   CACK
    INC     R1
    DJNZ    R3,  WRCCRDA                ;判断写完没有
RETWRNCCR:
    LCALL   STOP
    LCALL   DELAY10MS                   ;非易失性写周期10MS
    LCALL   DELAY10MS
    RET


;启动读实时时钟程序
;向X1205器件读秒/分/时/日/月/年/周/千年数据
;;秒/分/时/日/月/年/周/千年==>以CCR_DDR为首地址的连续8字节中
;占用：A、R1、R3、CY

READ_CCR:
    MOV     R3,  #08H                   ;
    LCALL   START                       ;启动总线
    MOV     A,   #0DEH                  ;X1205器件地址
    LCALL   WRBYTE                      ;发送器件从地址
    LCALL   CACK
    JNB     ACK,     RETRDNCCR          ;无应答则退出
    MOV     A,   #00H                   ;指定子地址高字节
    LCALL   WRBYTE
    LCALL   CACK
    MOV     A,   #30H                   ;指定子地址低字节(实时时钟地址）
    LCALL   WRBYTE
    LCALL   CACK

    LCALL   START                       ;重新启动总线
    MOV     A,   #0DFH
    LCALL   WRBYTE                      ;发读指令
    LCALL   CACK
    JNB     ACK,     READ_CCR
```

```
    MOV     R1,     #CCR_DDR        ;设置时间存储地址"40H"
RDN1:
    LCALL   RDBYTE                  ;读操作开始
    MOV     @R1,    A
    DJNZ    R3,     SACK
    LCALL   MNACK                   ;最后一字节法非应答信号
RETRDNCCR:
    LCALL   STOP
    LCALL   DELAY10MS
    RET
SACK:
    LCALL   MACK
    INC R1
    SJMP    RDN1


;---------------------------------------------
;(2) DELAY PROGRAM                |
;---------------------------------------------
DELAY10MS:
LCALL   DELAY5MS
LCALL   DELAY5MS
RET


DELAY5MS:
LCALL   DELAY1MS
LCALL   DELAY1MS
LCALL   DELAY1MS
LCALL   DELAY1MS
LCALL   DELAY1MS
RET

DELAY1MS:
    MOV TIME1, #02H
DELAY1MS1:
    MOV TIME0, #0D9H
DELAY1MS2:
    DJNZ TIME0, DELAY1MS2
    DJNZ TIME1, DELAY1MS1
RET
```

声明：本资料仅供参考。如有不同之处，请以相应英文资料为准。