

---

**ISD 1700 系列数码语音电路  
使用手册  
(中文版)**



[WWW.ATVOC.COM](http://WWW.ATVOC.COM)

## 前 言

ISD1700 系列是华邦公司新推出的语音芯片，用来替代已经停产的 ISD1400 系列及 ISD2500 系列芯片。ISD1700 系列不仅在录音时间上有更多的选择（从 20 秒到 240 秒），而且在功能上继承 14 及 25 系列的所有录放功能，并增加了一些更加人性化的提示功能及对存储地址的精确操作。根据我们经营 ISD 系列芯片多年的经验来看，ISD1700 的音质也较 14 及 25 系列有明显的提高。

本文根据华邦公司提供的 ISD1700 Design Guide Rev 1 做部分翻译，适用于 ISD1720, ISD1730, ISD1740, ISD1750, ISD1760, ISD1790, ISD17120, ISD17150 , ISD17180, ISD17210, ISD17240。

我们从来不标榜权威，但一直真诚的为大家服务，希望本手册可以帮助大家解决 ISD1700 使用上的问题及语音方案上的一些难题。您也可以登陆中青论坛 [www.atvoc.com/bbs](http://www.atvoc.com/bbs)，或许可以找到您要的答案，也可以和其他工程师做一些交流，得到更好的方案。

如有纰漏，欢迎批评指正！

# 目 录

|                            |    |
|----------------------------|----|
| 一、特点： .....                | 3  |
| 二、电特性： .....               | 3  |
| 三、独立按键工作模式.....            | 4  |
| 四 管脚功能说明： .....            | 8  |
| 五 ISD1730 的 SPI 模式.....    | 12 |
| 5.1 微机接口.....              | 12 |
| 5.2 SPI 协议总述.....          | 12 |
| 5.3 SPI 处理的格式.....         | 13 |
| 5.3.1 MOSI 数据格式.....       | 13 |
| 5.3.2 MISO 数据格式.....       | 15 |
| 5.4 SPI 协议格式.....          | 16 |
| 5.4.1 SPI 命令总览： .....      | 16 |
| 5.4.2 ISD1700 芯片内部寄存器..... | 17 |
| 六 ISD1700 的存储结构 .....      | 23 |
| 七 ISD1700S 典型应用电路.....     | 25 |

## ISD1700 系列多段语音录放芯片

ISD1700 系列芯片是华邦公司新推出的单片优质语音录放电路，该芯片提供多项新功能，包括内置专利的多信息管理系统，新信息提示(vAlert)，双运作模式（独立&嵌入式），以及可定制的信息操作指示音效。芯片内部包含有自动增益控制、麦克风前置放大器、扬声器驱动线路、振荡器与内存等的全方位整合系统功能。

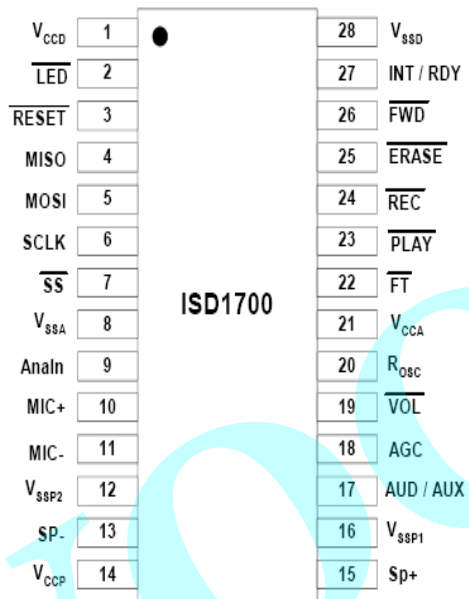
### 一、特点：

- 可录、放音十万次，存储内容可以断电保留一百年。
- 两种控制方式，两种录音输入方式，两种放音输出方式
- 可处理多达 255 段以上信息
- 有丰富多样的工作状态提示
- 多种采样频率对应多种录放时间
- 音质好，电压范围宽，应用灵活，价廉物美

### 二、电特性：

- 工作电压：2.4V-5.5V, ,最高不能超过 6V
- 静态电流：0.5 - 1  $\mu$ A
- 工作电流：20mA

用户可利用震荡电阻来自定芯片的采样频率，从而决定芯片的录放时



间和录放音质。下表为 ISD1700 的参数表：

| 采 样 率   | ISD1730 | ISD1740 | ISD1750  | ISD1760  | ISD1790  | ISD17120 | ISD17150 | ISD17180 | ISD17210 | ISD17240 |
|---------|---------|---------|----------|----------|----------|----------|----------|----------|----------|----------|
| 12 kHz  | 20 secs | 26 secs | 33 secs  | 40 secs  | 60 secs  | 80 secs  | 100 secs | 120 secs | 140 secs | 160 secs |
| 8 kHz   | 30 secs | 40 secs | 50 secs  | 60 secs  | 90 secs  | 120 secs | 150 secs | 180 secs | 210 secs | 240 secs |
| 6.4 kHz | 37 secs | 50 secs | 62 secs  | 75 secs  | 112 secs | 150 secs | 187 secs | 225 secs | 262 secs | 300 secs |
| 5.3 kHz | 45 secs | 60 secs | 75 secs  | 90 secs  | 135 secs | 181 secs | 226 secs | 271 secs | 317 secs | 362 secs |
| 4 kHz   | 60 secs | 80 secs | 100 secs | 120 secs | 180 secs | 240 secs | 300 secs | 360 secs | 420 secs | 480 secs |

而芯片的采样率可以通过外部振荡电阻来调节：

|         |               |               |                |                |                |
|---------|---------------|---------------|----------------|----------------|----------------|
| 采 样 频 率 | 12 kHz        | 8 kHz         | 6.4 kHz        | 5.3 kHz        | 4 kHz          |
| 振 荡 电 阻 | 60 k $\Omega$ | 80 k $\Omega$ | 100 k $\Omega$ | 120 k $\Omega$ | 160 k $\Omega$ |

### 三、独立按键工作模式

ISD1700 的独立按键工作模式录放电路非常简单（后附图），而且功能强大。不仅有录、放功能，还有快进、擦除、音量控制、直通放音和复位等功能。这些功能仅仅通过按键就可完成。

在按键模式工作时，芯片可以通过/LED 管脚给出信号来提示芯片的工作状态，并且伴随有提示音，用户也可自定 4 种提示音效。

#### ◇ 录音操作：

按下 REC 键，/REC 管脚电平变低后开始录音，直到松开按键使电平拉高或者芯片录满时结束。录音结束后，录音指针自动移向下一个有效地址。而放音指针则指向刚刚录完的那段语音地址。

#### ◇ 放音操作：

放音操作有两种模式，分别是边沿触发和电平触发，都由/PLAY 管脚触发。

*A) 边沿触发模式:*

点按一下 PLAY 键, /PLAY 管脚电平变低便开始播放当前段的语音, 并在遇到 EOM 标志后自动停止。放音结束后, 播放指针停留在刚播放的语音起始地址处, 再次点按放音键会重新播放刚才的语音。在放音期间, LED 灯会闪烁直到放音结束时熄灭。如果在放音期间点按放音键会停止放音。

*B) 电平放音模式:*

如果一直按住 PLAY 键, 使 /PLAY 管脚电平持续为低, 那么会将芯片内所有语音信息播放出来, 并且循环播放直到松开按键将 /PLAY 管脚电平拉高。在放音期间 LED 闪烁。当放音停止, 播放指针会停留在当前停止的语音段起始位置。

◇ 快进操作:

点按一下 FWD 按钮将 /FWD 端拉低, 会启动快进操作。快进操作用来将播放指针移向下一段语音信息。当播放指针到达最后一段语音处时, 再次快进, 指针会返回到第一段语音。当下降沿来到 /FWD 端时, 快进操作还要决定于芯片当时的状态:

*A)* 如果芯片在掉电状态并且当前播放指针的位置不在最后一段, 那么指针会前进一段, 到达下一段语音处。

*B)* 如果芯片在掉电状态并且当前播放指针的位置在最后一段, 那么指针会返回到第一段语音处。

*C)* 如果芯片正在播放一段语音 (非最后一段), 那么此时放音停止, 播放指针前进到下一段, 紧接着播放新的语音。

*D)* 如果芯片正在播放最一段语音, 那么此时, 放音停止, 播放指针返回到第一段语音, 紧接着播放第一段语音。

◇ 擦除操作:

擦除操作分为单段擦除和全体擦除两种擦除方式, 区别如下:

#### A) 单个擦除:

只有第一段或最后一段语音可以被单个擦除。点按一下 ERASE 键将 /ERASE 管脚拉低, 这时具体的擦除情况要看播放指针的状态:

- 如果芯片空闲并且播放指针指向第一段语音, 则会删除第一段语音, 播放指针指向新的第一段语音 (执行擦除操作前的第二段)
- 如果芯片空闲并且播放指针指向最后一段语音, 则会删除最后一段语音, 播放指针指向新的最后一段语音 (执行擦除操作前的倒数第二段)
- 如果芯片空闲并且播放指针指向没有指向第一或最后一段语音, 则不会删除任何语音, 播放指针也不会被改变
- 如果芯片当前正在播放第一段或最后一段语音, 点按下 ERASE 键会删除当前语音。

#### B) 全体擦除:

当按下 ERASE 键将 /ERASE 管脚电平拉低超过 2.5 秒钟, 会触发全体擦除操作, 删除全部语音信息。

##### ◇ 复位操作:

如果用 RESET 控制此管脚, 建议 /RESET 管脚与地之间连接一个  $0.1 \mu\text{F}$  电容。当 /RESET 被触发, 芯片将播放指针和录音指针都放置在最后一段语音信息的位置。

##### ◇ 音量操作:

点按一下 VOL 键将 /VOL 管脚拉低会改变音量大小。每按一下, 音量会减小一档, 再到达最小档后再按的话, 会增加音量直到最大档, 如此循环。总共有 8 个音量档供用户选择, 每一档会改变 4dB。复位操作会将音量档放在默认位置, 即最大音量。

##### ◇ FT 直通操作:

将 /FT 管脚与 GND 短接, 持续保持在低电平会启动直通模式。出厂设

定的是在芯片空闲状态，直通操作会将语音从Analn端直接通往喇叭端或AUD输出口。在录音期间开启FT功能，会同时录下Analn进入的语音信号。

◇ 提示音(SE)编辑:

ISD1700S 中设计了 4 种声音来提示当前的工作状态，分别为 SE1，SE2，SE3，SE4。

SE1: 录音，下一曲或全部擦除的开始;

SE2: 录音，单首擦除或最后一曲结束时;

SE3: 无效地擦除操作;

SE4: 全部擦除成功。

A) 进入SE 编辑模式:

1. 首先保持FWD 为低3 秒左右，然后LED 会闪一下（若有SE1，会同时播放SE1）。但是若当前曲目为最后一曲或没有录音则LED 会闪两下（若有SE2，会同时播放SE2）。

2. 保持FWD 为低，然后按下REC 使之为低直到LED 闪一下。

3. LED 再闪一下说明已经进入SE 编辑模式；进入此模式后，当前待编辑SE 为SE1。

B) 编辑:

进入SE 编辑模式后可按原来的方式进行录音，放音和擦除。按FWD



可选SE1 至SE4，按FWD 后可根据LED 的闪动次数来判断当前的SE，

闪一下为SE1，闪两下为SE2，依此类推。

A) 退出SE 编辑模式:

操作方法同进入方法一样。

SE 编辑的时间长度如下表所示:

|       |          |         |           |          |       |
|-------|----------|---------|-----------|----------|-------|
| 采样率   | 12KHz    | 8KHz    | 6.4KHz    | 5.3KHz   | 4KHz  |
| SE 时长 | 0.33 Sec | 0.5 Sec | 0.625 Sec | 0.75 Sec | 1 Sec |

四 管脚功能说明:

| 管脚名称   | PDIP/<br>SOIC<br>管脚 | TSOP<br>管脚 | 功 能   |
|--------|---------------------|------------|---|
| VCCD   | 1                   | 22         | 数字电路电源  |
| /LED   | 2                   | 23         | LED 指示信号输出  |
| /RESET | 3                   | 24         | 芯片复位  |
| MISO   | 4                   | 25         | SPI 接口的串行输出。ISD1700 在 SCLK 下降沿之前的半个周期将数据放置在 MISO 端。数据在 SCLK 的下降沿时移出 |

|       |    |    |  |
|-------|----|----|--|
| MOSI  | 5  | 26 | SPI 接口的数据输入端口。主控制芯片在 SCLK 上升沿之前的半个周期将数据放置在 MOSI 端。数据在 SCLK 上升沿被锁存在芯片内。此管脚在空闲时，应该被拉高  |
| SCLK  | 6  | 27 | SPI接口的时钟。由主控制芯片产生，并且被用来同步芯片MOSI和MISO端各自的数据输入和输出。此管脚空闲时，必须拉高。   |
| /SS   | 7  | 28 | 为低时，选择该芯片成为当前被控制设备并且开启 SPI 接口。空闲时，需要拉高   |
| VSSA  | 8  | 1  | 模拟地  |
| Analn | 9  | 2  | 芯片录音或直通时，辅助的模拟输入。需要一个交流耦合电容（典型值为0.1uF），并且输入信号的幅值不能超出1.0Vpp。APC寄存器的D3可以决定Analn信号被立刻录制到存储器中，与Mic信号混合被录制到存储器中，或者被缓存到喇叭端并经由直通线路从AUD/AUX输出。 |
| MIC+  | 10 | 3  | 麦克风输入+   |

|             |    |    |  |
|-------------|----|----|--|
| MIC-        | 11 | 4  | 麦克风输入-   |
| VSSP2       | 12 | 5  | 负极 PWM 喇叭驱动器地  |
| SP-         | 13 | 6  | 喇叭输出-  |
| VCCP        | 14 | 7  | PWM喇叭驱动器电源   |
| SP+         | 15 | 8  | 喇叭输出+  |
| VSSP1       | 16 | 9  | 正极 PWM 喇叭驱动器地  |
| AUD/<br>AUX | 17 | 10 | 辅助输出，决定于APC寄存器的D7，用来输出一个AUD或AUX输出。AUD是一个单端电流输出，而AuxOut是一个单端电压输出。他们能够被用来驱动一个外部扬声器。出厂默认设置为AUD。APC寄存器的D9可以使其掉电。 |
| AGC         | 18 | 11 | 自动增益控制   |
| /VOL        | 19 | 12 | 音量控制   |
| ROSC        | 20 | 13 | 振荡电阻ROSC用一个电阻连接到地，决定芯片的采样频率  |
| VCCA        | 21 | 14 | 模拟电路电源   |

|                 |    |    |   |
|-----------------|----|----|---|
| <b>/FT</b>      | 22 | 15 | 在独立芯片模式下，当 FT 一直为低，Analn 直通线路被激活。Analn 信号被立刻从 Analn 经由音量控制线路发射到喇叭以及 AUD/AUX 输出。D0 所控制。该管脚有一个内部上拉设备和一个内部防抖动设计，当在 SPI 模式下，SPI 无视这个输入，而且直通线路被 APC 寄存器的，允许使用按键开关来控制开始和结束。 |
| <b>/PLAY</b>    | 23 | 16 | 播放控制端   |
| <b>/REC</b>     | 24 | 17 | 录音控制端   |
| <b>/ERASE</b>   | 25 | 18 | 擦除控制端。  |
| <b>/FWD</b>     | 26 | 19 | 快进控制端   |
| <b>RDY /INT</b> | 27 | 20 | <p>一个开路输出。</p> <p><i>Ready(独立模式)</i></p> <p>该管脚在录音，放音，擦除和快进操作时保持为低，保持为高时进入空闲状态</p> <p><i>Interrupt(SPI 模式)</i></p> <p>在完成 SPI 命令后，会产生一个低信号的中断。一旦中断消除，该脚变回为高。</p>      |
| <b>VSSD</b>     | 28 | 21 | 数字地   |

## 五 ISD1730 的 SPI 模式

### 5.1 微机接口

主控单片机主要通过四线（SCLK, MOSI, MISO, /SS）SPI协议对ISD1700进行串行通信。ISD1700作为从机，几乎所有的操作都可以通过这个SPI协议来完成。为了兼容独立按键模式，一些SPI命令：PLAY, REC, ERASE, FWD, RESET和GLOBAL\_ERASE的运行类似于相应的独立按键模式的操作。另外，SET\_PLAY, SET\_REC, SET\_ERASE命令允许用户指定录音、放音和擦除的开始和结束地址。此外，还有一些命令可以访问APC寄存器，用来设置芯片模拟输入的方式。

### 5.2 SPI 协议总述

ISD1700系列的SPI串行接口操作遵照以下协议：

1. 一个SPI处理开始于/SS管脚的下降沿。
2. 在一个完整的SPI指令传输周期，/SS管脚必须保持低电平。
3. 数据在SCLK的上升沿锁存在芯片的MOSI管脚，在SCLK的下降沿从MISO管脚输出，并且首先移出低位。
4. SPI指令操作码包括命令字节，数据字节和地址字节，这决定于1700的指令类型
5. 当命令字及地址数据输入到MOSI管脚时，同时状态寄存器和当前行地址信息从MISO管脚移出。
6. 一个SPI处理在/SS变高后启动。
7. 在完成一个SPI命令的操作后，会启动一个中断信息，并且持续

保持为低，直到芯片收到CLR\_INT命令或者芯片复位。

### 5.3 SPI 处理的格式

图5.3.1 描述了SPI处理的格式。指令数据以数据队列的形式从MOSI移入芯片，第一个移入的字节是命令字节，这个字节决定了紧跟其后的数据类型。与此同时，芯片状态以及当前行地址信息以数据队列的方式通过MISO被返给主机。

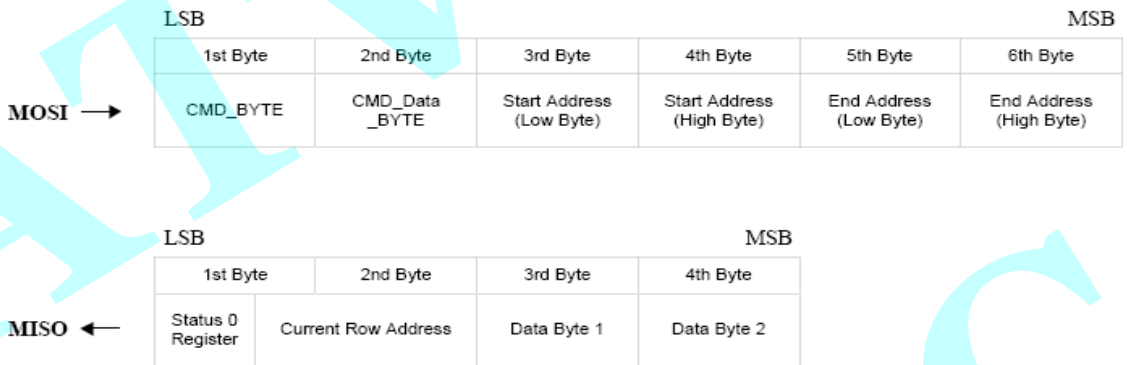


图5.3.1 SPI协议的数据格式

#### 5.3.1 MOSI 数据格式

MOSI是SPI接口的“主机输出从机接收端”。数据在SCLK的上升沿锁存进芯片，并且低位首先移出。1700的SPI指令格式依赖于命令的类型，根据不同类型的命令，指令可能是两个字节，也可能多达7个字节。MOSI的一般序列由下表列出，送到芯片的第一个字节是命令字节，这个字节确定了芯片将要完成的任务。其中命令字节的C4确定LED功能是否被激活。当C4=1，LED指示被开启，功能开启后，每一个SPI指令启动后，LED灯会闪亮一下。在命令字节之后，与之相关联的数据字节有可能包括对用来

存储信息进行精确操作的起始和结束地址。

| 1 <sup>st</sup> Byte: Command Byte                     |        |        |        |        |        |        |        |
|--|--------|--------|--------|--------|--------|--------|--------|
| MSB  |        |        |        |        |        |        | LSB    |
| Bit 7  | Bit 6  | Bit 5  | Bit 4  | Bit 3  | Bit 2  | Bit 1  | Bit 0  |
| C7   | C6     | C5     | C4     | C3     | C2     | C1     | C0     |
| 2 <sup>nd</sup> Byte: Data Byte1                       |        |        |        |        |        |        |        |
| MSB  |        |        |        |        |        |        | LSB    |
| Bit 15   | Bit 14 | Bit 13 | Bit 12 | Bit 11 | Bit 10 | Bit 9  | Bit 8  |
| X/D7   | X/D6   | X/D5   | X/D4   | X/D3   | X/D2   | X/D1   | X/D0   |
| 3 <sup>rd</sup> Byte: Data Byte2 / Start Address Byte1 |        |        |        |        |        |        |        |
| MSB  |        |        |        |        |        |        | LSB    |
| Bit 23   | Bit 22 | Bit 21 | Bit 20 | Bit 19 | Bit 18 | Bit 17 | Bit 16 |
| X/S7   | X/S6   | X/S5   | X/S4   | D11/S3 | D10/S2 | D9/S1  | D8/S0  |
| 4 <sup>th</sup> Byte: Data Byte3 / Start Address Byte2 |        |        |        |        |        |        |        |
| MSB  |        |        |        |        |        |        | LSB    |
| Bit 31   | Bit 30 | Bit 29 | Bit 28 | Bit 27 | Bit 26 | Bit 25 | Bit 24 |
| X  | X      | X      | X      | X      | S10    | S9     | S8     |
| 5 <sup>th</sup> Byte: End Address Byte1                |        |        |        |        |        |        |        |
| MSB  |        |        |        |        |        |        | LSB    |
| Bit 39   | Bit 38 | Bit 37 | Bit 36 | Bit 35 | Bit 34 | Bit 33 | Bit 32 |
| E7   | E6     | E5     | E4     | E3     | E2     | E1     | E0     |
| 6 <sup>th</sup> Byte: End Address Byte2                |        |        |        |        |        |        |        |
| MSB  |        |        |        |        |        |        | LSB    |
| Bit 47   | Bit 46 | Bit 45 | Bit 44 | Bit 43 | Bit 42 | Bit 41 | Bit 40 |
| X  | X      | X      | X      | X      | E10    | E9     | E8     |
| 7 <sup>th</sup> Byte: End Address Byte3                |        |        |        |        |        |        |        |
| MSB  |        |        |        |        |        |        | LSB    |
| Bit 55   | Bit 54 | Bit 53 | Bit 52 | Bit 51 | Bit 50 | Bit 49 | Bit 48 |
| X  | X      | X      | X      | X      | X      | X      | X      |

表 5.3.1 MOSI 数据顺序

多数的指令为两个字节，需要地址信息的指令则为七个字节。例如 LD\_APC 指令为三个字节，在其第二和第三字节是指令的数据字节。

有两种11位地址的设置，即 <S10:S0> 和 <E10:E0>，作为二进制地址的存放位置。芯片存储地址从第一个提示音的地址0X0000开始计算，但是 0x000-0x00F地址平均保留给了4个提示音。从0x010地址开始，才是非保留的存储区域，即真正的录音区。

### 5.3.2 MISO 数据格式

MISO即1700S的“主机接收从机发送”管脚，数据在SCLK的下降沿从MISO管脚输出，并且低位首先移出。对应每一个指令，MISO会伴随着指令码的输入，在前两个字节返回芯片当前的状态和行地址信息<A10:A0>。而RD\_STATUS, RD\_PLAY\_PNTR, RD\_REC\_PNTR 和 RD\_APC这些命令会在前两个字节之后产生额外的信息（见下边的细节说明）。

| 1 <sup>st</sup> Byte : Status Register 0 (Low Byte)   |                 |                    |                  |             |                   |            |            |
|---|-----------------|--------------------|------------------|-------------|-------------------|------------|------------|
| LSB   | Bit 0           | Bit 1              | Bit 2            | Bit 3       | Bit 4             | Bit 5      | MSB        |
|   | CMD Err         | Memory Full        | Power Up         | EOM         | Interrupt         | A0         | A2         |
| 2 <sup>nd</sup> Byte : Status Register 0 (High Byte)  |                 |                    |                  |             |                   |            |            |
| LSB   | Bit 8           | Bit 9              | Bit 10           | Bit 11      | Bit 12            | Bit 13     | MSB        |
|   | A3              | A4                 | A5               | A6          | A7                | A8         | A10        |
| 3 <sup>rd</sup> Byte : Data Byte 1 or SR0 (Low Byte)  |                 |                    |                  |             |                   |            |            |
| LSB   | Bit 16          | Bit 17             | Bit 18           | Bit 19      | Bit 20            | Bit 21     | MSB        |
|   | D0 /<br>CMD Err | D1/<br>Memory Full | D2 /<br>Power Up | D3 /<br>EOM | D4 /<br>Interrupt | D5 /<br>A0 | D7 /<br>A2 |
| 4 <sup>th</sup> Byte : Data Byte 2 or SR0 (High Byte) |                 |                    |                  |             |                   |            |            |
| LSB   | Bit 24          | Bit 25             | Bit 26           | Bit 27      | Bit 28            | Bit 29     | MSB        |
|   | D8 / A3         | D9 / A4            | D10 / A5         | D11 / A6    | D12 / A7          | D13 / A8   | D15 / A10  |
| 5 <sup>th</sup> Byte : SR0 (Low Byte)                 |                 |                    |                  |             |                   |            |            |
| LSB   | Bit 32          | Bit 33             | Bit 34           | Bit 35      | Bit 36            | Bit 37     | MSB        |
|   | CMD Err         | Memory Full        | Power Up         | EOM         | Interrupt         | A0         | A2         |
| 6 <sup>th</sup> Byte : SR0 (High Byte)                |                 |                    |                  |             |                   |            |            |
| LSB   | Bit 40          | Bit 41             | Bit 42           | Bit 43      | Bit 44            | Bit 45     | MSB        |
|   | A3              | A4                 | A5               | A6          | A7                | A8         | A10        |
| 7 <sup>th</sup> Byte : SR0 (Low Byte)                 |                 |                    |                  |             |                   |            |            |
| LSB   | Bit 48          | Bit 49             | Bit 50           | Bit 51      | Bit 52            | Bit 53     | MSB        |
|   | CMD Err         | Memory Full        | Power Up         | EOM         | Interrupt         | A0         | A2         |

表5.3.2 MISO数据时序

在输出信息中，第一字节的状态位提供了重要的信息，该信息标明了上一个SPI命令发送后的结果。例如，第一字节中的0位（command error bit）用来指示芯片是否接收了上一个SPI命令。而<A10:A0>地址位则给出了当前地址。第一和第二个数据字节的内容取决于上一个SPI命令。第五，第六和第七字节则是重复SR0状态寄存器的内容。



## 5.4 SPI 协议格式

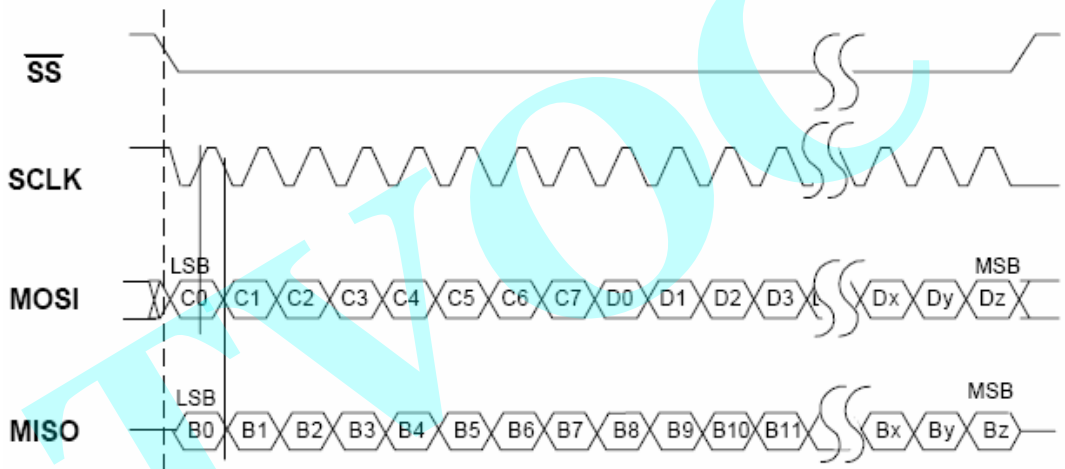


图5.4.1 SPI时序图

在SPI命令输入到ISD1700前，SPI端口的状态应该保持如下状态：

- ◎  $\overline{SS}$ =HIGH
- ◎ SCLK=HIGH
- ◎ MOSI=LOW

### 5.4.1 SPI 命令总览：

一个SPI命令总是由第一个命令字节开始。命令字节中的bit4位(LED)是具有特殊用途的。这个bit4位可以控制LED的输出。如果使用者想开启这个操作LED的功能，那么所有的SPI命令字都要将这个bit4位置1。

在SPI模式下，存储位置都可以通过行地址很容易地进行访问。主控单片机可以访问任何行地址，包括存储SE音效的行地址(0x000-0x00F)。像SET\_PLAY, SET\_REC和SET\_ERASE这些命令需要一个精确地起始地址和结束地址。如果开始地址和结束地址相同，那么ISD1700将只在这一行进行操作。SET\_ERASE操作可以精确地擦除在起始地址和结束地址间

的所有信息。SET\_REC操作从起始地址开始录音，并结束于结束地址，并且在结束地址自动加上EOM标志。同理，SET\_PLAY操作从起始地址播放语音信息，在结束地址停止播放。

另外，SET\_PLAY，SET\_REC和SET\_ERASE命令有一个先入先出的缓存器，使得从一个存储块到下一个存储块之间实现无缝转移。这个先入先出的缓存器只有在相同类型的SET命令下才有效。也就是说SET\_PLAY在SET\_ERASE之后将不能利用这个缓存器，并且这是一个错误的命令，SR0中的COM\_ERR位将被置1。当芯片准备好接收第二个SPI命令时，在SR1中的RDY位将置1。同样，在操作完成时会输出一个中断。例如，如果两个连续但带有两对不同地址的SET\_PLAY命令被正确发送后，此时缓存器装满。在完成第一个语音信息的播放后，第一个SET\_PLAY操作会遇到一个EOM，这时不会像一般遇到EOM时自动STOP，而是继续执行第二个SET\_PLAY命令，芯片将播放第二个语音信息。这个动作将最小化任何两个录音信息之间潜在的停留时间，且使芯片流畅地连接两个独立的信息。

如果循环存储体系处于令人满意的状态，那么可以使用PLAY，REC，FWD，RESET，ERASE和G\_ERASE这些命令，功能类似于1700的独立模式中相应的功能。这些命令将确保在独立模式下操作时储存机构保持一致，但是，音效提示将不同于独立模式。如果希望在SPI模式和独立模式之间转换，注意必须使用SET\_REC和SET\_ERASE以遵循循环存储体系。

#### 5.4.2 ISD1700 芯片内部寄存器

这些寄存器可以返回1700芯片的内部状态，下边将描述每个寄存器的数据模式：

### Status Register0 (SR0)

状态寄存器SR0是两字节数据，由MISO返回。它包括5个状态位（D4：D0）以及11个地址位（A10：A0）

| SR0          |         | Size:  | 16 bits |     |     | Type: | Read |      |         |  |
|--------------|---------|--|---------|-----|-----|-------|------|------|---------|--|
| Byte #1      | Bit # : | D7   | D6      | D5  | D4  | D3    | D2   | D1   | D0      |  |
|              | Name :  | A2   | A1      | A0  | INT | EOM   | PU   | FULL | CMD_ERR |  |
| Byte #2      | Bit # : | D15  | D14     | D13 | D12 | D11   | D10  | D9   | D8      |  |
|              | Name :  | A10  | A9      | A8  | A7  | A6    | A5   | A4   | A3      |  |
| Description: |         | Device status register                                   |         |     |     |       |      |      |         |  |
| Access       |         | Every SPI command returns SR0 as first two bytes in MISO |         |     |     |       |      |      |         |  |

表5.4.1 SR0格式

| SR0  |   |         |   |
|------|---|---------|---|
|      | 位 | 名称      | 描述  |
| 第一字节 | 7 | A2      | 当前行地址bit2   |
|      | 6 | A1      | 当前行地址bit1   |
|      | 5 | A0      | 当前行地址bit0   |
|      | 4 | INT     | 当一个操作完成时，该位置1，可被CLR_INT命令清除                           |
|      | 3 | EOM     | 当检测到EOM时此位置1，可被CLR_INT命令清除                            |
|      | 2 | PU      | 当芯片在SPI模式中上电后，此位置1                                    |
|      | 1 | FULL    | 此位置1表示存储区已满。芯片不能再录任何信息，除非旧的信息被删除。此位只有遵循独立按键模式的存储体系才有效 |
|      | 0 | CMD_ERR | 此位置1表示上一个SPI命令无效                                      |

|      |    |     |            |
|------|----|-----|------------|
| 第二字节 | 15 | A10 | 当前行地址bit10 |
|      | 14 | A9  | 当前行地址bit9  |
|      | 13 | A8  | 当前行地址bit8  |
|      | 12 | A7  | 当前行地址bit7  |
|      | 11 | A6  | 当前行地址bit6  |
|      | 10 | A5  | 当前行地址bit5  |
|      | 9  | A4  | 当前行地址bit4  |
|      | 8  | A3  | 当前行地址bit3  |

表5.4.2 SR0内容说明

### Status Register1 (SR1)

| SR1 |       |  |
|-----|-------|--|
| 位   | 名称    | 描述   |
| 7   | SE1   | 当音效1被录入后此位置1, 擦除后置0  |
| 6   | SE2   | 当音效1被录入后此位置1, 擦除后置0  |
| 5   | SE3   | 当音效1被录入后此位置1, 擦除后置0  |
| 4   | SE4   | 当音效1被录入后此位置1, 擦除后置0  |
| 3   | REC   | 此位置1表示正在录音   |
| 2   | PLAY  | 此位置1表示正在放音   |
| 1   | ERASE | 此位置1表示正在擦除   |
| 0   | RDY   | RDY=1时表示空闲, 可以接收SPI命令;RDY=0时为忙, 不接受新的命令, 但除了RESET, CLR_INT, RD_STATUS, PD这些指令, 当然在REC和PLAY时, 也可以接受STOP命令。当 |

|  |  |                         |
|--|--|-------------------------|
|  |  | 其他命令发送时，会被忽略且CMD_ERR被置1 |
|--|--|-------------------------|

表5.4.3 SR1内容说明

APC(模拟通道寄存器)

| 位       | 名称                        | 描述                             | 默认值          |   |            |
|---------|---------------------------|--------------------------------|--------------|---|------------|
| D0      | VOL0                      | 音量控制：<br>000=最大<br>111=最小      | 000          |   |            |
| D1      | VOL1                      |                                |              |   |            |
| D2      | VOL2                      |                                |              |   |            |
| D3      | Monitor_Input             | 录音时输入监听信号<br>0=关闭监听<br>1=开启监听  | 0            |   |            |
| D4      | Mix_Input                 | 混合输入。                          |              | 0 |            |
|         |                           | D4=0                           | FT/D6=0      |   | 线录录音       |
|         |                           |                                | FT/D6=1      |   | MIC录音      |
|         |                           | D4=1                           | FT/D6=0      |   | MIC与线录混合录音 |
| FT/D6=1 | MIC录音                     |                                |              |   |            |
| D5      | SE_Editing                | 0=开启音效编辑，<br>1=关闭音效编辑          | 0            |   |            |
| D6      | SPI_FT                    | SPI模式下，由此位控制直通功能               |              | 1 |            |
|         |                           | D6=0                           | 直通功能SPI模式下开启 |   |            |
|         |                           | D6=1                           | 直通功能SPI模式下关闭 |   |            |
| D7      | Analog Output:<br>AUD/AUX | 选择AUD输出或AUX输出：<br>0=AUD, 1=AUX | 0            |   |            |

|            |                      |   |          |
|------------|----------------------|---|----------|
| <b>D8</b>  | <b>PWM SPK</b>       | <b>PWM输出：0=开启；1=关闭</b>  | <b>0</b> |
| <b>D9</b>  | <b>PU Analog Out</b> | <b>电源模拟输出：<br/>0=ON<br/>1=OFF</b>                             | <b>0</b> |
| <b>D10</b> | <b>VAlert</b>        | <b>VAlert: 0=ON, 1=OFF</b>                                    | <b>1</b> |
| <b>D11</b> | <b>EOM Enable</b>    | <b>此位置1时，用指定地址放音（SETPLAY），在遇到EOM标志时，放音停止。置0时，则放音持续到直到结束地址</b> | <b>0</b> |

表5.4.4 APC内容说明

### 5.4.3 1700SPI指令表

| 指令名称             | 命令字节        | 数据字节1       | 数据字节2<br>或起始<br>地址字节1 | 数据字节3<br>或起始<br>地址字节2 | 结束地址<br>字节1/2/3 | 描述                       |
|------------------|-------------|-------------|-----------------------|-----------------------|-----------------|--------------------------|
| <b>PU</b>        | <b>0x01</b> | <b>0x00</b> |                       |                       |                 | <b>1700上电</b>            |
| <b>STOP</b>      | <b>0x02</b> | <b>0x00</b> |                       |                       |                 | <b>停止当前操作</b>            |
| <b>RESET</b>     | <b>0x03</b> | <b>0x00</b> |                       |                       |                 | <b>芯片复位</b>              |
| <b>CLR_INT</b>   | <b>0x04</b> | <b>0x00</b> |                       |                       |                 | <b>清除EOM标志和中<br/>断信号</b> |
| <b>RD_STATUS</b> | <b>0x05</b> | <b>0x00</b> | <b>0x00</b>           |                       |                 | <b>返回状态寄存器内<br/>容</b>    |

|                    |             |                      |                                |             |  |  |
|--------------------|-------------|----------------------|--------------------------------|-------------|--|--|
| <b>RD_PLAY_PTR</b> | <b>0x06</b> | <b>0x00</b>          | <b>0x00</b>                    | <b>0x00</b> |  | 在1, 2字节返回状态寄存器信息, 2, 3字节返回放音指针         |
| <b>PD</b>          | <b>0x07</b> | <b>0x00</b>          |                                |             |  | 芯片下电                                   |
| <b>RD_REC_PTR</b>  | <b>0x08</b> | <b>0x00</b>          | <b>0x00</b>                    | <b>0x00</b> |  | 在1, 2字节返回状态寄存器信息, 2, 3字节返回录音指针         |
| <b>DEVID</b>       | <b>0x09</b> | <b>0x00</b>          | <b>0x00</b>                    |             |  | 读取芯片ID号在                               |
| <b>PLAY</b>        | <b>0x40</b> | <b>0x00</b>          |                                |             |  | 从当前地址放音                                |
| <b>REC</b>         | <b>0x41</b> | <b>0x00</b>          |                                |             |  | 从当前地址录音                                |
| <b>ERASE</b>       | <b>0x42</b> | <b>0x00</b>          |                                |             |  | 擦除当前段语音                                |
| <b>G_ERASE</b>     | <b>0x43</b> | <b>0x00</b>          |                                |             |  | 擦除芯片所有语音                               |
| <b>RD_APC</b>      | <b>0x44</b> | <b>0x00</b>          | <b>0x00</b>                    | <b>0x00</b> |  | 返回状态寄存器信息在1, 2字节, APC寄存器信息在 3字节        |
| <b>WR_APC1</b>     | <b>0x45</b> | <b>&lt;D7:D0&gt;</b> | <b>&lt;xxxx<br/>D11:D8&gt;</b> |             |  | 将 <D11:D0> 写入 APC寄存器                   |
| <b>WR_APC2</b>     | <b>0x65</b> | <b>&lt;D7:D0&gt;</b> |                                |             |  | 将 <D11:D0> 写入 APC寄存器, 并由 <D2:D0> 位控制音量 |
| <b>WR_NVCFG</b>    | <b>0x46</b> | <b>0x00</b>          |                                |             |  | 将 APC 内容写入 NVCFG                       |

|           |          |      |         |                  |                                |                               |
|-----------|----------|------|---------|------------------|--------------------------------|-------------------------------|
| LD_NVCFG  | 0x47     | 0x00 |         |                  |                                | 将NVCFG内容写入APC                 |
| FWD       | 0x48     | 0x00 |         |                  |                                | 将放音指针指向下一段语音起始地址              |
| CHK_MEM   | 0x49     | 0x00 |         |                  |                                | 检查环状存储体系                      |
| EXTCLK    | 0x4<br>A | 0x00 |         |                  |                                | 外部时钟使能                        |
| SET_PLAY  | 0x80     | 0x00 | <S7:S0> | <xxxx<br>S10:S8> | <xxxx xxxx<br>xxxxxE10:<br>E0> | 从起始地址放音<S10:S0>到结束地址E10:E0>停止 |
| SET_REC   | 0x81     | 0x00 | <S7:S0> | <xxxx<br>S10:S8> | <xxxx xxxx<br>xxxxxE10:<br>E0> | 从起始地址录音<S10:S0>到结束地址E10:E0>停止 |
| SET_ERASE | 0x82     | 0x00 | <S7:S0> | <xxxx<br>S10:S8> | <xxxx xxxx<br>xxxxxE10:<br>E0> | 从起始地址擦除<S10:S0>到结束地址E10:E0>停止 |

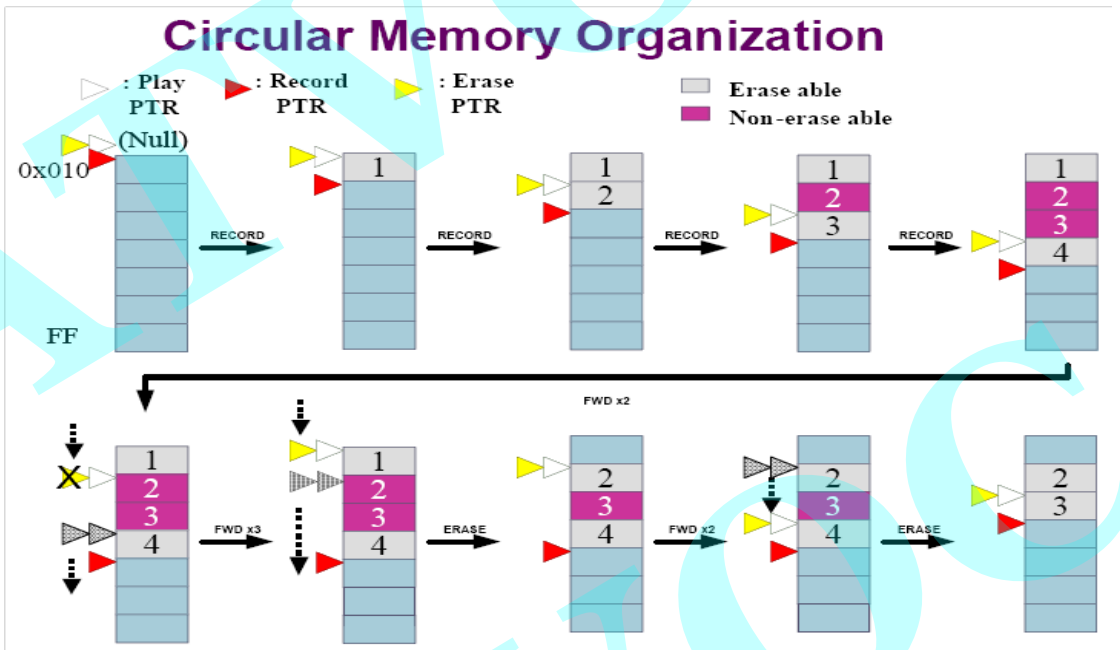
## 六 ISD1700 的存储结构

在独立按键模式下，芯片内有一套环形存储结构管理系统来管理录音段的存放。当芯片读写存储器时会检查是否合法的存储结构，若不是则LED会闪7下，然后芯片将不接受任何指令除了复位和全部擦除指令。遇到这种情况需先将芯片成功全部擦除才能复原，这样原来的内容将全部丢失除了提示音。



环形存储结构管理系统管理的地址是0x10 到末地址, 0x00 至0x0f 为 SE 的地址。当地址指针指到末地址后, 会自动跳到0x10。在此管理系统下录音段之间是连续存放的, 但首地址与末地址之间至少有一个空地址间隔来让系统区分首末地址。

在SPI 模式下, 用户可对任意地址进行操作, 但若不按管理系统的方式存储或擦除录音段, 在独立按键模式下将不能操作。



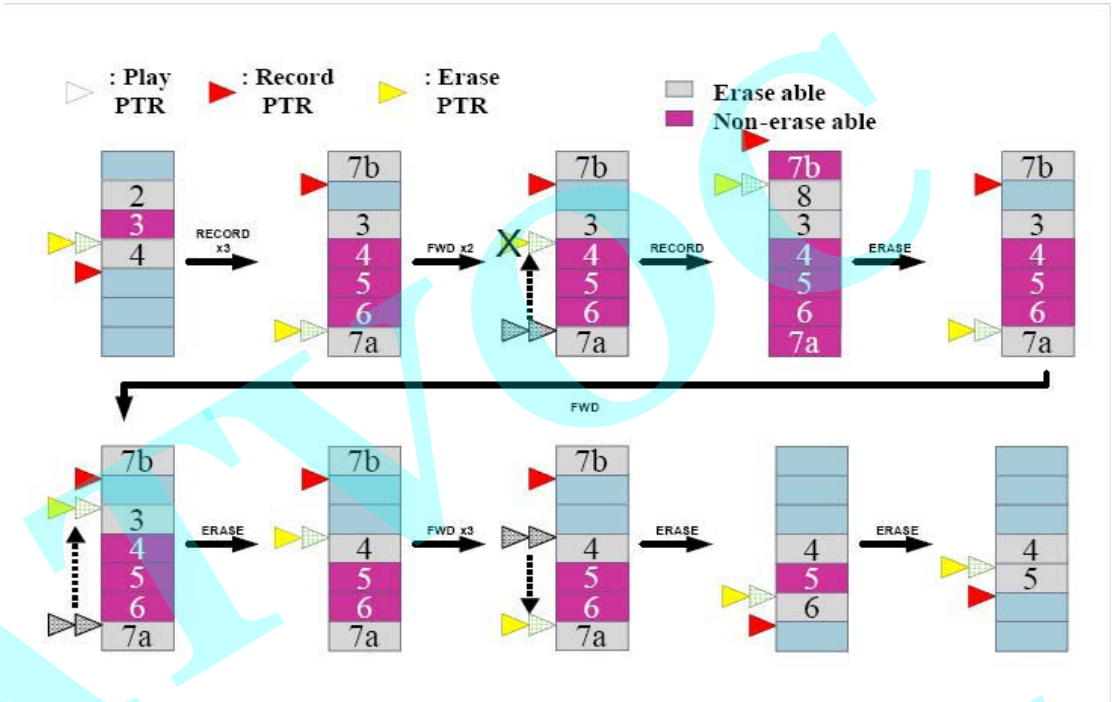


图6.0.0 1700存储体系示意图

### 七 ISD1700S 典型应用电路

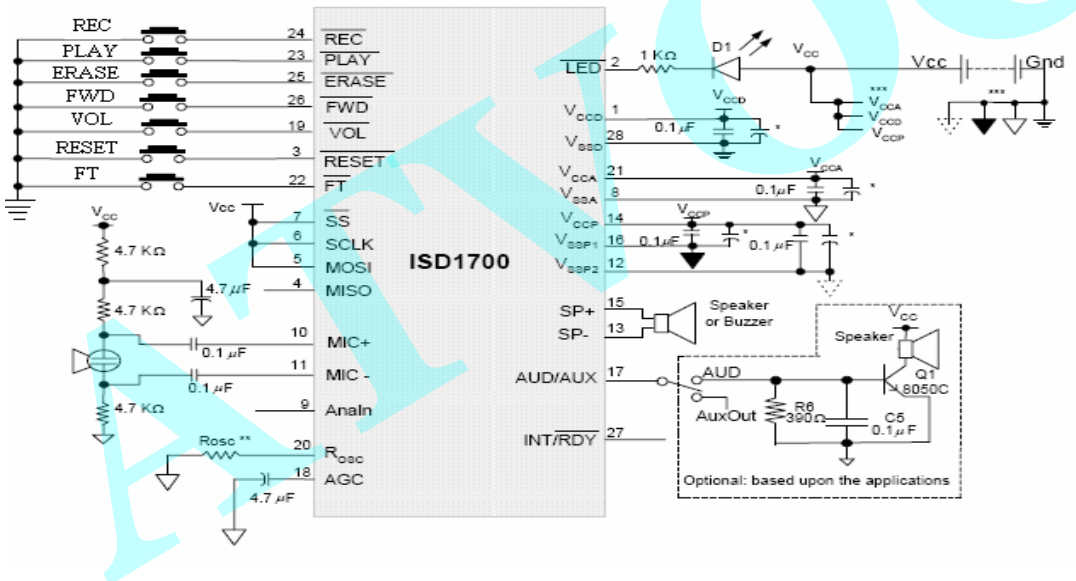


图 1 MIC 录音电路

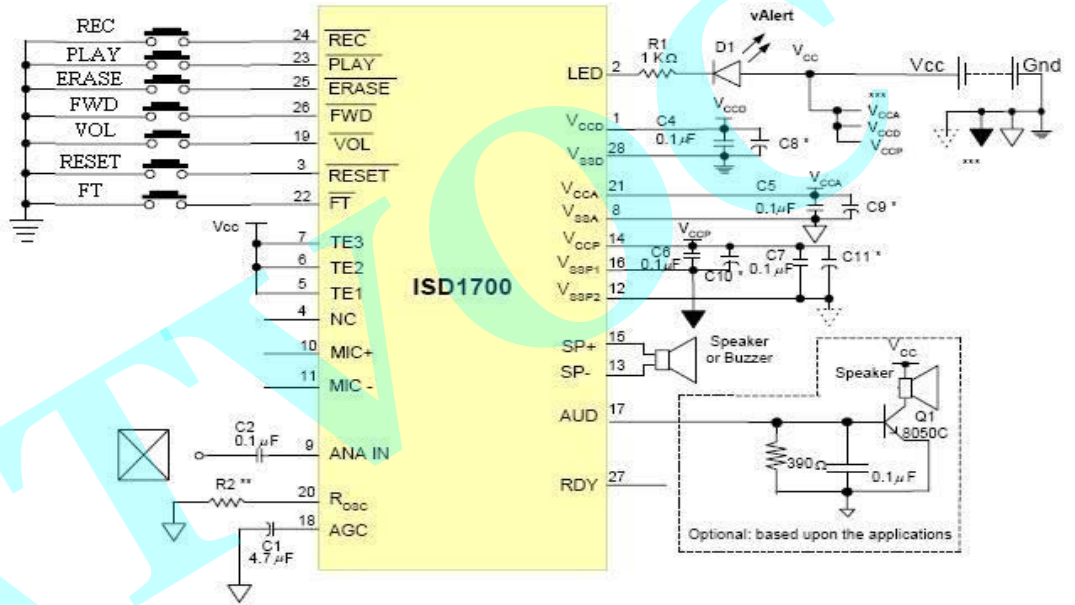


图 2 AnalIn 输入电路

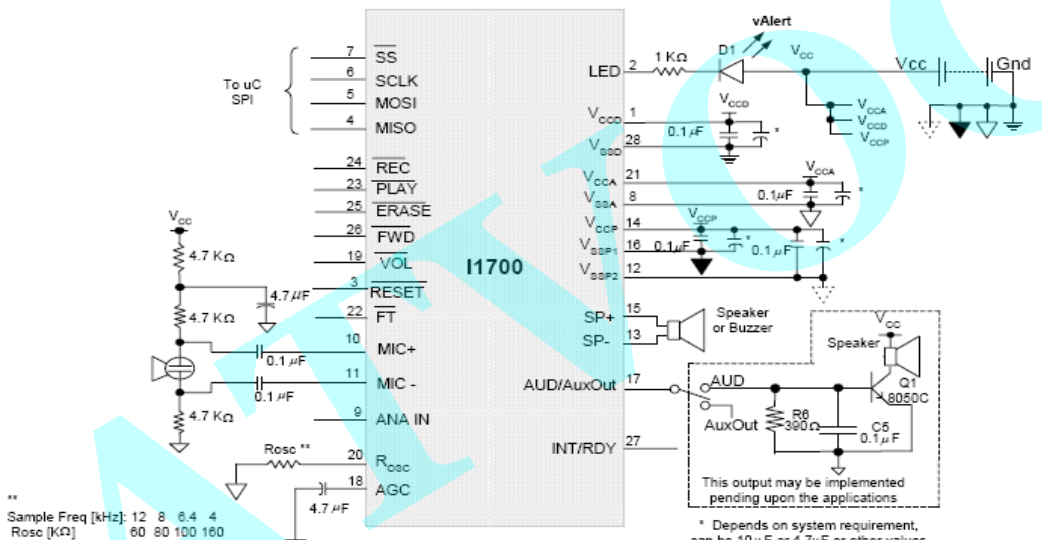


图 3 SPI 接口控制电路