# ISD ChipCorder®
# ISD2100 Series
# Design Guide



For additional information or questions, please contact: Nuvoton Technology Corporation.

www.nuvoton.com

**TABLE OF CONTENTS**

# 1 GENERAL DESCRIPTION

The ISD2100 is a digital ChipCorder® providing single-chip storage and playback of high quality audio. The device features digital de-compression, comprehensive memory management, flash storage, and integrated audio signal path and Class D speaker driver capable of delivering power of 400mW. This family utilizes flash memory to provide non-volatile audio playback with duration up to 30 seconds (based on 8kHz/4bit ADPCM compression) for a single-chip audio playback solution.

The ISD2100 can be controlled and programmed through an SPI serial interface or operated stand-alone by triggers applied to the device's six GPIO pins.

The ISD2100 requires no external clock sources or components except a speaker to deliver quality audio prompts or sound effects to enhance user interfaces.

In addition, the part can provide non-volatile flash storage in 1Kbyte sectors eliminating the need for additional serial EEPROM/Flash devices.

ISD2100 provides wide range of sampling frequencies, high SNR performance, low power consumption, fast programming time and integrated program verification.

# 2 FEATURES

- Duration
  - **ISD2130 –** 30 seconds based on 8kHz/4bit ADPCM in 1Mbit of flash storage
  - **ISD2115A** – 15 seconds based on 8kHz/4bit ADPCM in 512Kbit of flash storage
- Audio Management
  - Store pre-recorded audio (**Voice Prompts**) using high quality digital compression
  - Use simple index based command for playback – no address needed.
  - Execute pre-programmed macro scripts (**Voice Macros**) designed to control the configuration of the device and playback Voice Prompts sequences.
- Control
  - Serial SPI interface for microprocessor control and programming.
  - Stand-alone control where customized Voice Macro scripts are assigned to GPIO trigger pins.
- Sample Rate
  - 7 sampling frequencies 4, 5.3, 6.4, 8, 12.8, 16 and 32 kHz are available.
  - Each Voice Prompt can have optimal sample rate.
- Compression Algorithms
  - µ-Law: 6, 7 or 8 bits per sample
  - Differential µ-Law: 6, 7 or 8 bits per sample
  - PCM: 8, 10 or 12 bits per sample
  - Enhanced ADPCM: 2, 3, 4 or 5 bits per sample
  - Variable-bit-rate optimized compression. This allows best possible compression given a metric of SNR and background noise levels.
- Oscillator
  - Internal oscillator with internal reference: factory trimmed to ±1% deviation at room temperature.

- Output
  - PWM: Class D speaker driver to direct drive an 8Ω speaker or buzzer.
  - Delivers:
    - 4Ω load: 500mW @3V; 630mW @3.3V.
    - 8Ω load: 330mW @3V; 420mW @3.3V.
- I/Os
  - SPI interface: MISO, MOSI, SCLK, SSB for commands and digital audio data
  - 6 general purpose I/O pins multiplexed with SPI interface.
- Flash Storage
  - 1Mbit (ISD2130) or 512Kbit (ISD2115A) of storage for combined audio/data.
  - Fast programming time (20µs/byte)
  - Erase sector size 1Kbyte, sector erase time 2ms.
  - Integrated memory checksum calculation for fast verification.
  - Endurance >100K cycles. Retention > 10 years
- Operating Voltage: 2.7-3.6V
- Package: green, 20L-QFN, 14L-SOP
- Temperature Options:
  - Industrial: -40°C to 85°C

## 3 BLOCK DIAGRAM

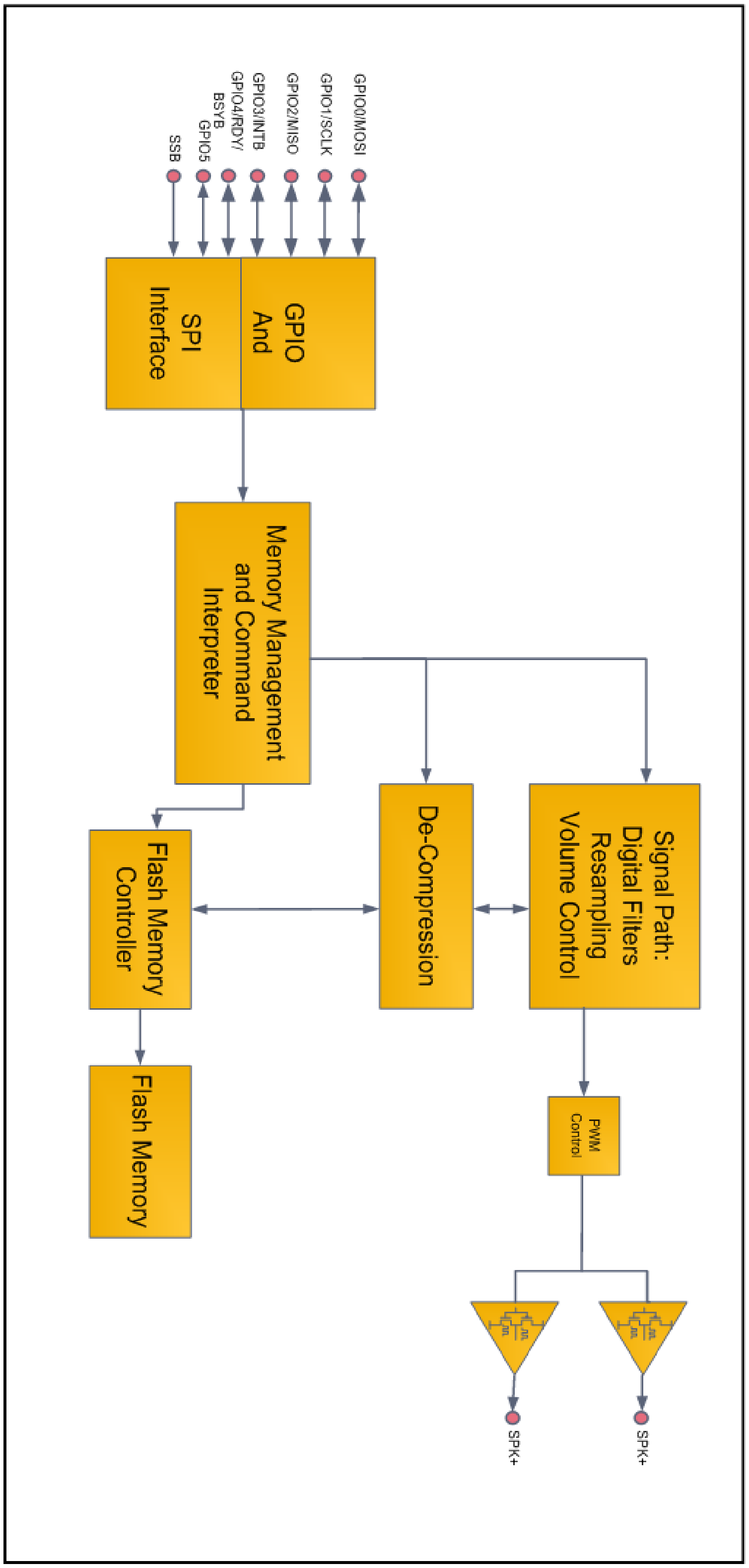



Figure 3-1 ISD2100 Block Diagram

**4 BLOCK DIAGRAM**

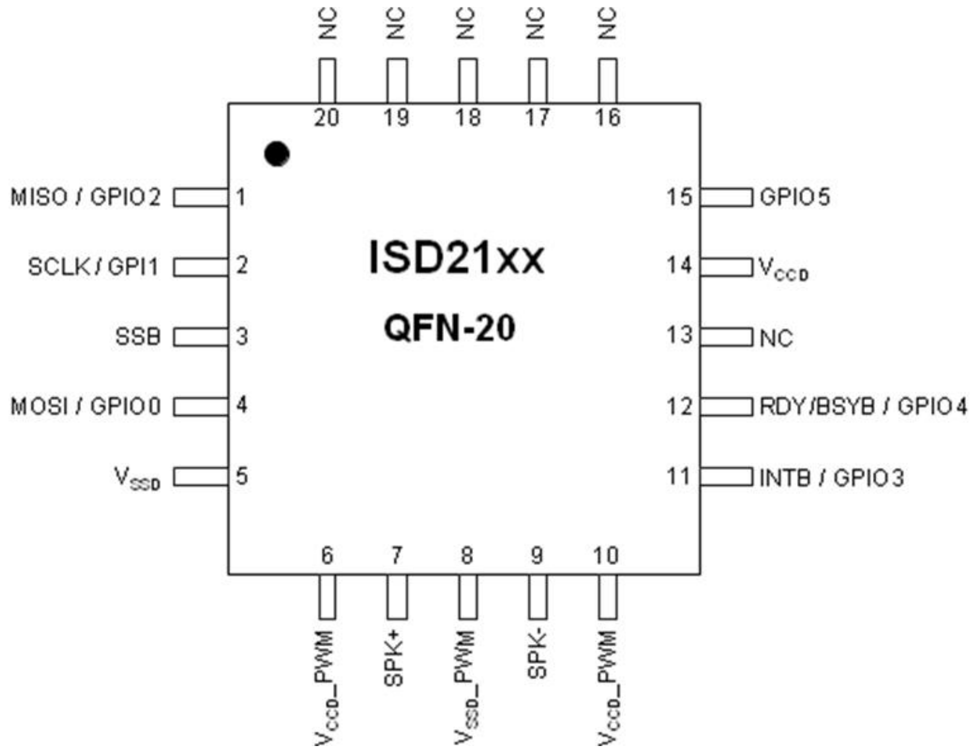



Figure 4-1 ISD2100 20-Lead QFN Pin Configuration.

*Note: the center exposed pad underneath of the QFN20 package is electrically connected to back of the die substrate with GROUND potential; it should be connected to VSSD.

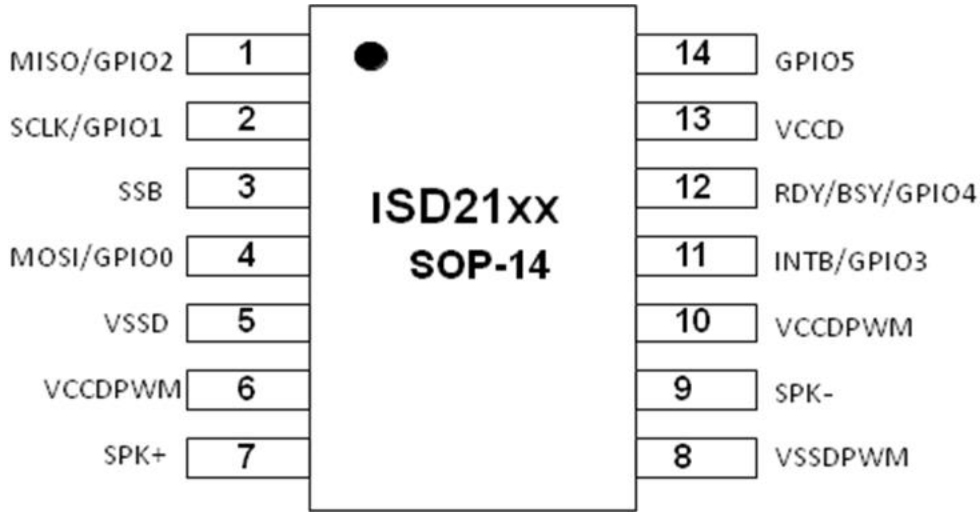



Figure 4-2 ISD2100 SOP14 Pin Configuration

## 5 PIN DESCRIPTION

### ISD2100 QFN20 Pin Configuration

| Pin # | Pin Name | I/O | Function |
|---|---|---|---|
| 1 | MISO / GPIO2 | O | Master-In-Slave-Out. Serial output from the ISD2100 to the host. This pin is in tri-state when SSB=1.<br>Can be configured as a general purpose I/O pin. |
| 2 | SCLK / GPI1 | I | Serial Clock input to the ISD2100 from the host.<br>Can be configured as a general purpose input pin. |
| 3 | SSB | I | Slave Select input to the ISD2100 from the host. When SSB is low device is selected and responds to commands on the SPI interface. When asserted, GPIO0/1/2 automatically configure to MOSI/SCLK and MISO respectively. SSB has an internal pull-up to Vccd. |
| 4 | MOSI / GPIO0 | I | Master-Out-Slave-In. Serial input to the ISD2100 from the host.<br>Can be configured as a general purpose I/O pin. |
| 5 | $V_{SSD}$ | I | Digital Ground. |
| 6 | $V_{CCD}$_PWM | I | Digital Power for the PWM Driver. It can be from a separate different power supply other than $V_{CCD}$. |
| 7 | SPK+ | O | PWM driver positive output. This SPK+ output, together with SPK- pin, provide a differential output to drive 8Ω speaker or buzzer. During power down this pin is in tri-state. |
| 8 | $V_{SSD}$_PWM | I | Digital Ground for the PWM Driver. |
| 9 | SPK- | O | PWM driver negative output. This SPK- output, together with SPK+ pin, provides a differential output to drive 8Ω speaker or buzzer. During power down this pin is tri-state. |
| 10 | $V_{CCD}$_PWM | I | Digital Power for the PWM Driver. It can be from a separate different power supply other than $V_{CCD}$. |
| 11 | INTB / GPIO3 | O | Active low interrupt request pin. This pin is an open-drain output.<br>Can be configured as a general purpose I/O pin. |
| 12 | RDY/BSYB / GPIO4 | O | An output pin to report the status of data transfer on the SPI interface. “High” indicates that ISD2100 is ready to accept new SPI commands or data. Can be configured as a general purpose I/O pin. |
| 13 | NC | | This pin should be left unconnected. |
| 14 | $V_{CCD}$ | I | Digital Power. . It can be from a separate different power supply other than $V_{CCD}$_PWM. |
| 15 | GPIO5 | I/O | General purpose I/O pin |
| 16 | NC | | This pin should be left unconnected. |
| 17 | NC | | This pin should be left unconnected. |
| 18 | NC | | This pin should be left unconnected. |
| 19 | NC | | This pin should be left unconnected. |

| Pin # | Pin Name | I/O | Function |
|---|---|---|---|
| 20 | NC | | This pin should be left unconnected. |

### ISD2100 SOP14 Pin Configuration

| Pin # | Pin Name | I/O | Function |
|---|---|---|---|
| 1 | MISO / GPIO2 | O | Master-In-Slave-Out. Serial output from the ISD2100 to the host. This pin is in tri-state when SSB=1.<br>Can be configured as a general purpose I/O pin. |
| 2 | SCLK / GPI1 | I | Serial Clock input to the ISD2100 from the host.<br>Can be configured as a general purpose input pin. |
| 3 | SSB | I | Slave Select input to the ISD2100 from the host. When SSB is low device is selected and responds to commands on the SPI interface. When asserted, GPIO0/1/2 automatically configure to MOSI/SCLK and MISO respectively. SSB has an internal pull-up to Vccd. |
| 4 | MOSI / GPIO0 | I | Master-Out-Slave-In. Serial input to the ISD2100 from the host.<br>Can be configured as a general purpose I/O pin. |
| 5 | $V_{SSD}$ | I | Digital Ground. |
| 6 | $V_{CCD}$_PWM | I | Digital Power for the PWM Driver. |
| 7 | SPK+ | O | PWM driver positive output. This SPK+ output, together with SPK- pin, provide a differential output to drive 8Ω speaker or buzzer. During power down this pin is in tri-state. |
| 8 | $V_{SSD}$_PWM | I | Digital Ground for the PWM Driver. |
| 9 | SPK- | O | PWM driver negative output. This SPK- output, together with SPK+ pin, provides a differential output to drive 8Ω speaker or buzzer. During power down this pin is tri-state. |
| 10 | $V_{CCD}$_PWM | I | Digital Power for the PWM Driver. |
| 11 | INTB / GPIO3 | O | Active low interrupt request pin. This pin is an open-drain output.<br>Can be configured as a general purpose I/O pin. |
| 12 | RDY/BSYB / GPIO4 | O | An output pin to report the status of data transfer on the SPI interface. “High” indicates that ISD2100 is ready to accept new SPI commands or data. Can be configured as a general purpose I/O pin. |
| 13 | $V_{CCD}$ | I | Digital Power. |
| 14 | GPIO5 | I/O | General purpose I/O pin |

# 6 DEVICE OPERATION

Playback of audio stored on the ISD2100 can be accomplished by either sending SPI commands via the serial interface or triggered by signal edges applied to GPIO pins. The device is programmed via the SPI interface either in-system or utilizing commercially available gang programmers.

## 6.1 AUDIO STORAGE

The audio compression and customization of the ISD2100 is rapidly achieved with the supplied ISD2100VPE or Voice Prompt Editor. This software tool allows the developer to take audio clips in standard wave file format and re-sample and compress them for download to the ISD2100.

Audio is stored in the ISD2100 as series of **Voice Prompts**: these units of audio can be of any length – the compression and sample rate of each Voice Prompt can be individually selected. A powerful feature of the ISD2100 is presence of a scripting ability **Voice Macros**. A Voice Macro can contain commands to play individual Voice Prompts and configure the ISD2100. A Voice Macro can be associated with a GPIO pin such that it is triggered by a transition on that pin. In this way stand-alone systems can be developed without the need for micro-controller interaction. Voice Macros can also be executed via the SPI command interface. Both Voice Prompts and Voice Macros are addressed via a simple sequential index address, no absolute memory address is required, thus audio source material or voice macro function can be updated (or changed for multi-language implementation) without the need to update microcontroller code.

## 6.2 DEVICE CONFIGURATION

The ISD2100 is configured by writing to a set of configuration registers. This can be accomplished either by sending configuration via the serial SPI interface or executing Voice Macros containing configuration commands. Most configuration registers are reset to their default values when the device is powered down to ensure lowest possible standby current. Exceptions to this are registers that control the configuration of GPIO pins and Jump registers that contain the Voice Macro index to execute for GPIO triggers. Configuration registers may be initialized automatically in customizable Voice Macros that are executed on a power-on reset or power-up condition. Section 12 contains a complete list of all configuration registers in the device.

### 6.3 GPIO CONFIGURATION

The six GPIO pins of the ISD2100 can be configured for a variety of purposes. Each pin can be configured to trigger a Voice Macro function. Each pin also has an alternate function allowing the pins to be configured as SPI, interrupt or oscillator reference pins.

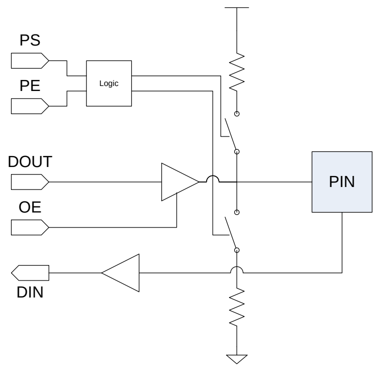



Figure 6-1 GPIO Structure

The structure of the GPIO pads is shown in Figure 6-1. Configuration registers allow the user to control pull-up and pull down resistors, enable the pin as an output or set the output value. See Section 12.3 for details on the configuration options. The AF0 and AF1 bits associated with each pin determine its functionality as shown in Table 6-1. When the GPIO is selected for its alternate function, the DOUT and OE connections to the pin are driven by sources other than the DOUT and OE registers. The alternate functions of each pin are summarized in Table 6-2. In addition, the Slave Select Bar pin (SSB) when driven active low will force GPIO0-2 into alternate function mode to allow an SPI transaction. Note that the pull selects (PS) and pull enable (PE) bits continue to function in alternate function mode – it is possible to connect or disconnect pull devices to the pin even in alternate function mode.

AF[1:0] settings 0b10 and 0b11 associate the GPIO with Voice Macro triggers. Setting 0b10 triggers a Voice Macro when a falling edge occurs on the GPIO. Setting 0b11 triggers a voice macro on both the falling and rising edges of the GPIO. These triggers are de-bounced internally and are thus suitable for use with push-buttons. See Section 10 for a complete description of GPIO triggers.

Table 6-1 Alternate Function Bit Decoding

| AF1,AF0 | **Configuration** |
|---|---|
| 00 | **GPIO** |
| 01 | **Alternate Function** |
| 10 | **GPIO, falling edge trigger** |
| 11 | **GPIO, falling and rising edge trigger** |

Table 6-2 GPIO Alternate Configurations and effect on DOUT and OE.

| GPIO | **Alternate Function** | **DOUT** | **OE** |
|---|---|---|---|
| GPIO0 | **MOSI** | **X** | **L** |
| GPIO1 | **SCLK** | **X** | **L**[1] |
| GPIO2 | **MISO** | **MISO** | **SS**[2] |
| GPIO3 | **INTB** | **L** | **INT**[3] |
| GPIO4 | **RDY/BSY** | **RDY** | **H** |
| GPIO5 | **REXT** | **L** | **L** |

### 6.4 Oscillator and Sample Rates

The ISD2100 has an internal oscillator trimmed at manufacturing that requires no external components to operate. This oscillator provides an internal clock source that operates the ISD2100 at a maximum audio sample rate $F_{Smax}$ of 32kHz. The sample rates available for audio storage at this maximum sample rate are shown in Table 6-3. The sample rate is selected during compression using the ISD2100 Voice Prompt Editor software.

Table 6-3 Available Sample Rates.

| SR[2:0] | Ratio to $F_{Smax}$ | Sample Rate $F_S$ (kHz) |
|---|---|---|
| 0 | 8 | 4 |
| 1 | 6 | 5.44 |
| 2 | 5 | 6.4 |
| 3 | 4 | 8 |
| 4 | 2.5 | 12.8 |
| 5 | 2 | 16 |
| 6 | 1 | 32 |

[1] GPIO1 OE is permanently low to guarantee SPI operation. GPIO1 cannot be used as an output.

[2] SS=!SSB. MISO is tri-state for SSB=1 and driven for SSB=0.

[3] GPIO3 is INTB pin. INTB is normally tri-state and driven low for interrupt.

## 7 MEMORY FORMAT

The memory of the ISD2100 consists of byte addressable flash memory that is erasable in 1Kbyte sectors. Erased memory has a value of 0xFF. Writing to the memory allows host to change bits from erased '1' state to programmed '0' state.

The memory of the ISD2100 is organized into four distinct regions as shown in Figure 7-1. The four regions are:

1. **Configuration and Index Table**: The first region of memory contains configuration data for the device and the index table that points to the Voice Prompt and Voice Macro data. The ISD2100VPE creates this section for download to the device.
2. **Voice Macros**: This section contains the script code of all the projects Voice Macros.
3. **Voice Prompts**: This section contains the compressed audio data for all Voice Prompts.
4. **User Data**: An optional section containing memory sectors allocated by the developer for generic use by the host controller.

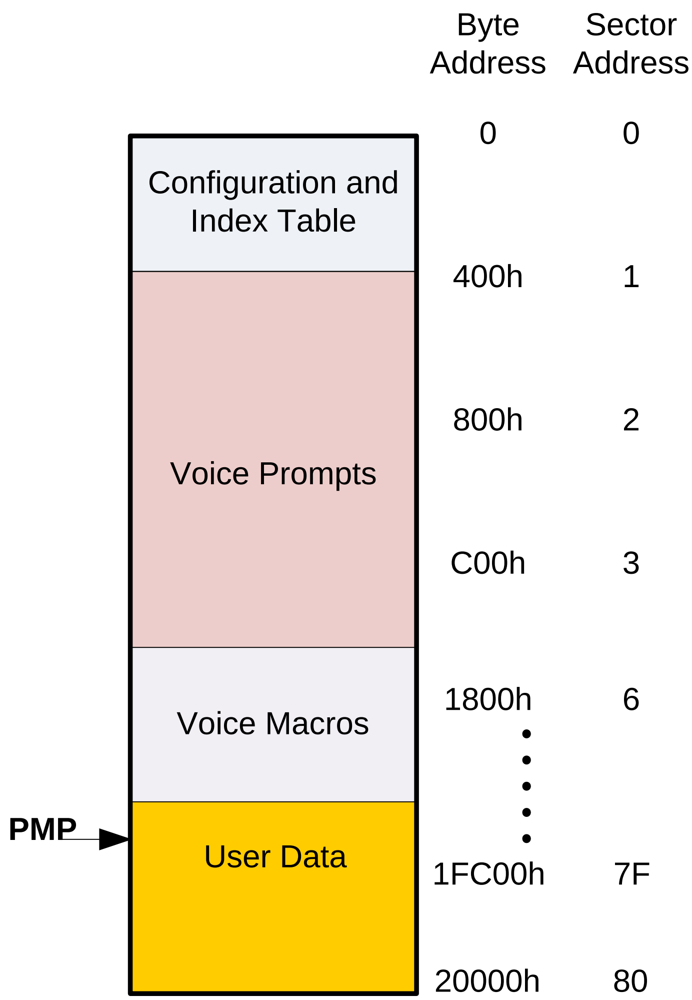



Figure 7-1 ISD2100 Memory Organization

#### 7.1.1 Voice Prompts

Voice prompts are pre-recorded audio of any length, from short words, phrases or sound effects to long passages of music. These Voice Prompts can be played back in any order as determined by the application. A Voice Prompt consists of two components:

1. An index entry in the Index Table pointing to the pre-recorded audio.
2. Compressed pre-recorded audio data.

A Voice Prompt is addressed using its index number to locate and play the pre-recorded audio. This address free approach allows users to easily manage the pre-recorded audio without the need to update the code on the host controller. In addition, the users can store a multitude of pre-recorded audio without the overhead of maintaining a complicated lookup table. To assist customers in creating the Voice Prompts, ISD2100 Voice Prompt Editor and writer are available for development purposes.

#### 7.1.2 Voice Macros

Voice Macros are a script that allows users to customize their own play patterns such as play Voice Prompts, insert silence, power-down the device and configure the signal path, including volume control. Voice Macros are executed using a single SPI command and are accessed using the same index structure as Voice Prompts. This means that a Voice Macro (or Voice Prompt) can be updated on the ISD2100 without the need to update code on the host micro-controller since absolute addresses are not needed.

The following locations have been reserved for special Voice Macros:

Index 0: Power-On Initialization (POI)

Index 1: Power-Up (PU)

Index 2: GPIO-Wakeup (WAKEUP)

These Voice Macros allow the users to customize the ISD2100 power-on, power-up and GPIO wake-up procedures and are executed automatically when utilized. If these Voice Macros are not used device will perform default operations on these events.

An example to illustrate the usage of the PU Voice Macro is:

- WR_CFG(VOLC, 0x0C) ; Set VOLC to 0x0C
- WR_CFG(REG2, 0x44) ; Set REG2 to 0x44
- WR_CFG (REG_GPIO_AF1 ,0xFF); Set REG_GPIO_AF1 to 0xFF
- WR_CFG (REG_GPIO_AF0 ,0x10) ; Set REG_GPIO_AF0 to 0x10
- FINISH ; Exit Voice Macro

The above PU Voice Macro will perform the following:

- Choose Volume Control for -3dB level.
- Configure and power up the signal path to decode compressed audio to speaker driver.
- Set up all GPIOs except GPIO4 for Falling edge trigger and set GPIO4 for both falling and rising edge trigger.

The following is the complete list of the command available for use in Voice Macros:

- WR_CFG_REG(*reg n*) – Set configuration register *reg* to value *n*.
- PWR_DN – Power down the ISD2100.
- PLAY_VP(*i*) – Play Voice Prompt index *i*.
- PLAY_VP@(*Rn*) – Indirect Play Voice Prompt of index in register *Rn*
- PLAY_VP_LP(*i,cnt*) – Loop Play Voice Prompt index *i, cnt* times.
- PLAY_VP_LP@(*Rn,cnt*) – Indirect Loop Play Voice Prompt index in *Rn, cnt* times.
- EXE_VM(*i*) – Execute Voice Macro index *i*.
- EXE_VM@(*Rn)* – Indirect Execute Voice Macro index in register *Rn*
- PLAY_SIL(*n*) – Play silence for *n* units. A unit is 32ms at master sampling rate of 32 kHz.
- WAIT_INT – Wait until current play command finishes before executing next macro instruction.
- FINISH – Finish the voice macro and exit.

These commands are equivalent to the commands available via the SPI interface and are described in Section 13.

### 7.1.3 User Data

User Data consists of 1KByte multiples of erasable sectors allocated by the user. This can be used as generic non-volatile storage by the host application. The developer has the freedom not to allocate or reserve any memory sectors. A software tool, the ISD2100 Voice Prompt Editor is available to assist customers in allocating such memory.

### 7.2 MEMORY HEADER

The memory header is located from address 0x000000 and is used to determine the format and function of the ISD2100. The Memory Header stores users' configurable information including the memory protection scheme, the protection (PMP) pointer and the index table including POI, PU and other Voice Macros defined by the users. The ISD2100 Voice Prompt Editor software creates the memory header for the developer, the description here is for reference.

Table 7-1 Memory Header

| Initial Bytes of the Memory Header | | | | | | |
|---|---|---|---|---|---|---|
| **Byte0** | **Byte1-2** | **Byte3-4** | **Bytes5-10** | **Bytes11-16** | **Byte17-22** | **Byte23-28** |
| 0xCX | - | PMP[15:0] | POI_VM | PU_VM | GPIO-Wakeup | VM/VP[3] |

The Memory Header contains at least seventeen bytes located at the beginning of the memory space. Byte0 determines the memory protection scheme. Bytes 3-4 store the PMP pointer. Byte 5 onwards is the Voice Prompt/Voice Macro index table. This table consists of six byte entries that are start and end address of Voice Prompts or Voice Macros. Bytes 5-10 & Bytes11-16 are reserved for the POI and PU Voice Macros which are the first two entries in this table (index 0 and index 1) to be executed on power-on initialization and power-up, respectively. Bytes 17-22 are reserved for GPIO-Wakeup Voice Macro. If these functions are not desired, these entries should be left erased (0xFFFFFF,0xFFFFFF). When a PLAY_VP(*i*) or EXE_VM(*i*) command is sent to the ISD2100, it reads the index table entry at address 6*i*+5 and executes the VP or VM at the address present in the table.

The PMP points to the boundary of protected memory and is used in conjunction with the RP, WP and CEP bits to set memory protection indicated below (also see Section **Error! Reference source not found.** for details).

Table 7-2 Memory Header Byte0

| Memory Header Byte | | | | | | | |
|---|---|---|---|---|---|---|---|
| **Bit 7** | **Bit 6** | **Bit 5** | **Bit 4** | **Bit 3** | **Bit 2** | **Bit 1** | **Bit 0** |
| 1 | 1 | 0 | 0 | 1 | RP | WP | CEP |

### 7.3 Digital Access of Memory

The ISD2100 memory can be accessed as conventional digital memory using the DIG_READ, DIG_WRTIE, ERASE_MEM and CHIP_ERASE SPI commands. This allows the user to:

- Use areas of memory as digital non-volatile memory.
- Update Voice Prompts and macros (pre-recorded audio) in system.
- Read and verify Voice Prompt memory.
- Erase memory in 1Kbyte sectors.
- Mass Erase entire memory.

The digital memory commands can only be issued while an audio playback is inactive. The RDY/BSYB pin governs the flow control for all digital operations. See Section 13 for details on the digital memory commands.

### 7.4 Memory Contents Protection

Under certain circumstances, it is desirable to protect portions of the internal memory from write/erase or interrogation (read). The ISD2100 provides a method to achieve this by setting a protection memory pointer (PMP) that allows the users to protect memory for an address range from the beginning of memory to this sector containing the PMP pointer. The type of protection is set by three bits in the memory header byte.

- The **CEP** (Chip Erase Protect) bit set to zero enables chip erase protection. This prevents a mass erase function, allowing the device to be configured as a write-once part. With the **CEP** bit set to one, even with write protection enabled, the part can be mass erased. After mass erasure, the initial sector byte defaults to no protection so the device can be re-programmed.
- The **WP** (Write Protect) bit set to zero enables write protection of the internal memory below the sector pointed to by the PMP. Write protection means that digital write or erase commands will not function in this memory area. This can be used to ensure that audio or other data is not inadvertently erased or overwritten.
- The **RP** (Read Protect) bit set to zero enables read protection of the internal memory below the sector pointed to by the PMP. Read protection means that digital read or audio playback commands through SPI will not function in this memory area. This can be used to ensure that memory contents cannot be digitally copied or read.

Memory protection is activated on power-up of the chip. Therefore, each time the user changes the setting of memory protection, the new setting will not be effective until the chip is reset.

## 8 SPI INTERFACE

This is a standard four-wire serial interface used for communication between ISD2100 and the host. It consists of an active low slave-select (SSB), a serial clock (SCLK), a data input (Master Out Slave In - MOSI), and a data output (Master In Slave Out - MISO). In addition, for some transactions requiring data flow control, a RDY/BSYB signal (pin) is available.

The ISD2100 supports **SPI mode 3:** (1) SCLK must be high when SPI bus is inactive, and (2) data is sampled at SCLK rising edge. A SPI transaction begins on the falling edge of SSB and its waveform is illustrated below:

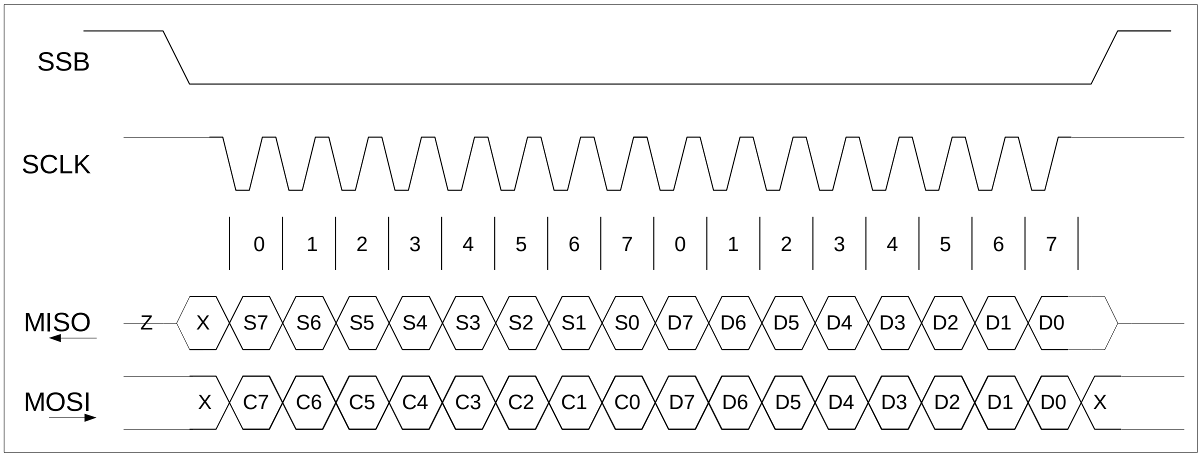



Figure 8-1 SPI Data Transaction.

A transaction begins with sending a command byte (C7-C0) with the most significant bit (MSB – C7) sent in first. During the byte transmission, the status (S7-S0) of the device is sent out via the MISO pin. After the byte transmission, depending upon the command sent, one or more bytes of data will be sent via the MISO pin.

RDY/BSYB pin is used to handshake data into or out of the device. Upon completion of a byte transmission, RDY/BSYB pin could change its state after the rising edge of the SCLK if the built-in 32-byte data buffer is either full or empty. At this point, SCLK must remain high until RDY/BSYB pin returns to high, indicating that the ISD2100 is ready for the next data transmission. See below for timing diagram.

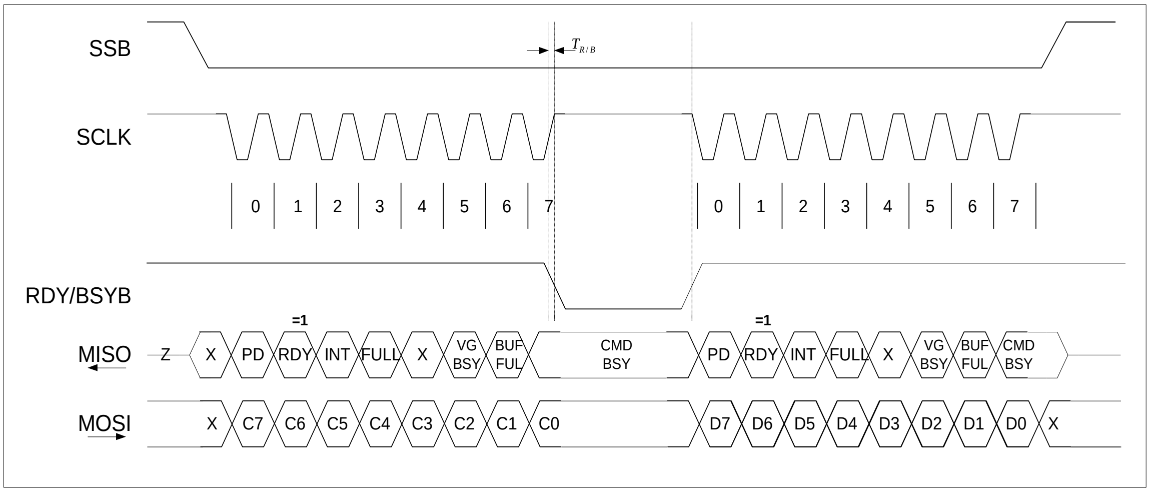



Figure 8-2 RDY/BSYB Timing for SPI Writing Transactions.

If the SCLK does not remain high, RDY bit of the status register will be set to zero and be reported via the MISO pin so the host can take the necessary actions (i.e., terminate SPI transmission and re-transmit the data when the RDY/BSYB pin returns to high).

For commands (i.e., DIG_READ, SPI_PCM_READ) that read data from the ISD2100 device, MISO is used to read the data; therefore, the host must monitor the status via the RDY/BSYB pin and take the necessary actions. The INT pin will go low to indicate (1) data overrun/overflow when sending data to the ISD2100; or (2) invalid data from ISD2100. See Figure 8-3 for the timing diagram.

To avoid RDY/BSYB polling for digital operations the following conditions must be met:

- Ensure device is idle (CMD_BSY=0 in status) before operation.
- Digital Write: Send 32 bytes of data or less in a digital write transaction **or** ensure that there is a 24µs period between each byte sent where SCLK is held high.
- Digital Read: Ensure a 2µs period between last address byte of digital read command and first data byte where SCLK is held high.

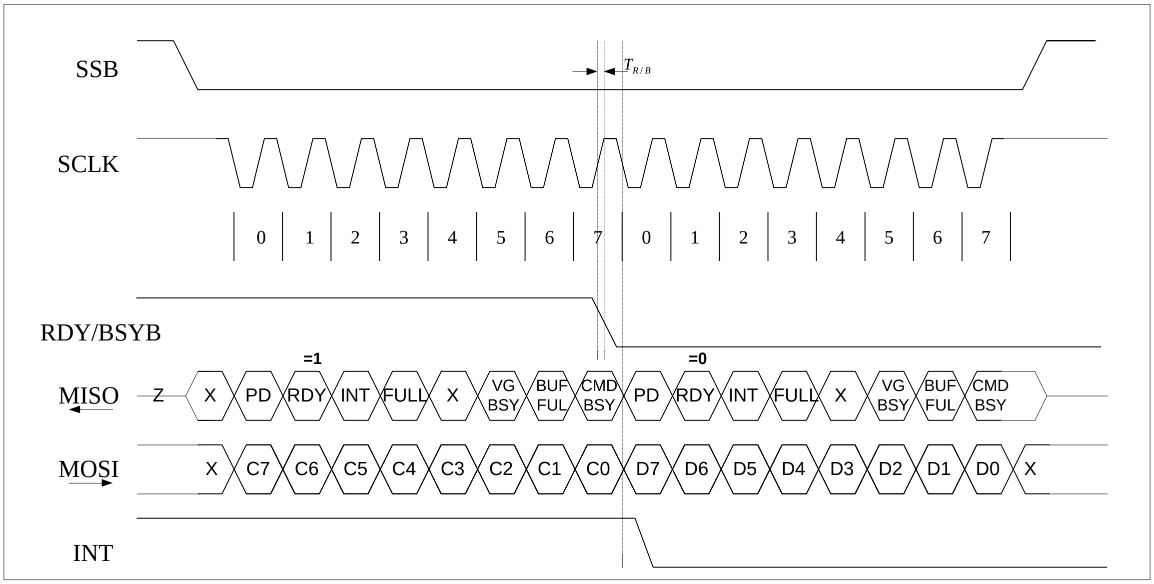



Figure 8-3 SPI Transaction Ignoring RDY/BSYB

## 9 SIGNAL PATH

The signal path performs filtering, sample rate conversion, volume control and decompression. A block diagram of the signal path is shown in Figure 9-1. The PWM driver output pins SPK- and SPK+ provide a differential output to drive an 8Ω speaker or buzzer. During power down these pins are in tri-state.

Pre-compressed audio transfers from memory or SPI interface through the de-compressor block to PWM driver or SPI out. The audio level is adjustable via VOLC before going out on to the PWM driver path. The possible path combinations are:

MEMORY → DECOMPRESS → SPKR (Playback to speaker)

MEMORY → DECOMPRESS → SPI_OUT (SPI playback)

SPI_IN → DECOMPRESS → SPKR (SPI decode to speaker)

For example, to playback audio to speaker, enable decompression and PWM (write 0x44 to register 0x02) then send a PLAY_VP command to play audio.

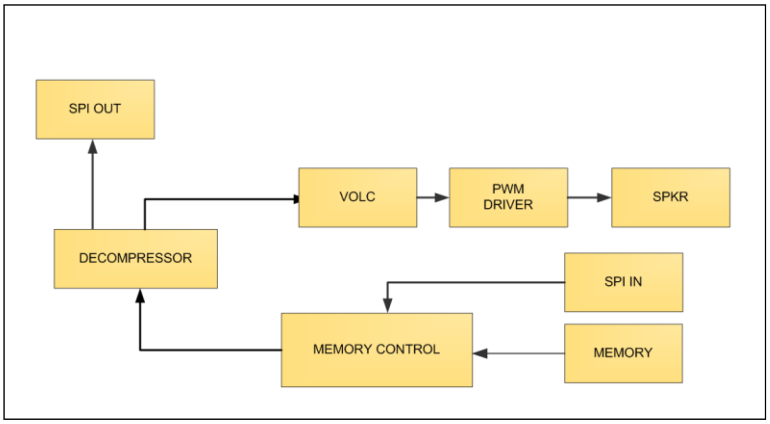



Figure 9-1 ISD2100 Signal Path

## 10 GPIO VOICE MACRO TRIGGERS

The ISD2100 Voice Macro capability and GPIO flexibility allows the user to configure the device to operate independently of the SPI interface or host micro-controller.

GPIO triggering utilizes the Jump registers R0 through R6. When a GPIO trigger event occurs the ISD2100 executes the Voice Macro whose index is stored in the corresponding Jump register: that is GPIO0 will execute the VM whose index is stored in R0, GPIO1 in R1 etc. The initial values of the R0-R6 registers can be set up in the POI Voice macro which is executed when a power-on reset condition is detected. When the ISD2100 responds to a trigger event, if a Voice Macro is currently being executed, that Voice Macro is first stopped before execution of new Voice Macro.

### 10.1 Voice Macro Examples

Below are some useful examples demonstrating the features Voice trigger macros. The example project can be found in the ISD2100VPE distribution as the ISD2100example project.

#### 10.1.1 POI/PU/WAKEUP Voice Macros

These special purpose Voice Macros allow the user to configure the ISD2100 for subsequent trigger events. The POI macro is executed when the chip receives an internal power-on reset condition or the SPI SW_RESET command is sent.

The POI Voice macro is used to configure the ISD2100 for subsequent trigger events, for example:

a. CFG(REG2, 0x44) ; Configure signal path to playback
b. CFG(VOLC, 0x00) ; Set Volume to 0dB
c. CFG(R5, 0x03) ; Set Jump register R5 to 0x03, GPIO5 to trigger VM#3
d. CFG(R4, 0x07) ; Set Jump register R4 to 0x07, GPIO4 to trigger VM#7
e. CFG(R3, 0x09) ; Set Jump register R3 to 0x09, GPIO3 to trigger VM#9
f. CFG(R2, 0x0a) ; Set Jump register R2 to 0x0a, GPIO2 to trigger VM#A
g. CFG(R1, 0x0c) ; Set Jump register R1 to 0x0c, GPIO1 to trigger VM#C
h. CFG(R0, 0x0e) ; Set Jump register R0 to 0x0e, GPIO0 to trigger VM#E
i. PLAY_VP(FastBeep) ; Play Voice Prompt FastBeep
j. CFG(REG_GPIO_AF1, 0xff) ; Set up GPIOs to trigger off falling edges
k. CFG(REG_GPIO_AF0, 0x00)
l. PD ; Power Down

This POI macro will initialize the GPIO configuration such that all GPIO triggers are enabled for falling edges and performs initialization of the jump registers to point to appropriate Voice Macros. It also configures the play path and plays a beep. At the end of the macro the chip powers down.

The GPIO_WAKEUP is executed whenever the device is triggered from a power down state.

a. CFG(REG2, 0x44) ; Configure signal path to playback

b. CFG(VOLC, 0x00) ; Set Volume to 0dB
c. CFG(R4, 0x07) ; Set Jump register R4 to 0x07, GPIO4 to trigger VM#7
d. CFG(R2, 0x0a) ; Set Jump register R2 to 0x0a, GPIO2 to trigger VM#A
e. Finish ; Exit Voice Macro, stay powered up.

This GPIO_WAKEUP macro sets up the play path as settings in these registers are reset during power down. It also resets jump registers R4 and R2 to default conditions.

#### 10.1.2 Example: Cycle through a sequence of messages.

In this example a high-to-low transition on GPIO5 will initially trigger VM#3 as defined in the POI initialization macro. In VM#3 the Voice Prompt “One” is played and jump register R5 set to VM#4. Thus the next high-to-low transition on GPIO5 will trigger VM#4 and play Voice Prompt “Two”. Similarly next trigger will play “Three” then “Four” and back to “One”. Notice the difference in VM#4 where a WAIT_INTERRUPT command has been inserted before the setting of the jump register. If the GPIO5/SW6 button is pushed rapidly, so that play is interrupted, “Two” will continue to be repeated. Other Voice Macros, because the jump register is changed first, will always progress to the next step in sequence.

- **VM#3**: **R5_Count_One (GPIO5)**
  a. CFG(R5, 0x04) ; Configure GPIO5 to play VM#4 on next trigger
  b. Play(One) ; Play voice prompt “One”
  c. PD ; Power Down
- **VM#4:Two**
  a. Play(Two) ; Play voice prompt “Two”
  b. Wait Interrupt ; Wait until Play finishes
  c. CFG(R5, 0x05) ; Configure GPIO5 to play VM#5 on next trigger
  d. PD ; Power Down
- **VM#5: Three**
  a. CFG(R5, 0x06) ; Configure GPIO5 to play VM#6 on next trigger
  b. Play(Three) ; Play voice prompt “Three”
  c. PD ; Power Down
- **VM#6: Four**
  a. CFG(R5, 0x03) ; Configure GPIO5 to play VM# 3 on next trigger
  b. Play(Four) ; Play voice prompt “Four “
  c. PD ; Power Down

### 10.1.3 Example: Looping short sounds. Interrupt to stop playback.

This example demonstrates how to loop short sound samples and use a trigger interrupt to stop playback. A trigger on GPIO4 will play a series of Voice Prompts until it is interrupted by another trigger to stop playback. VM#7 was associated with the GPIO4 trigger in the POI routine. The first action of this VM is to change the trigger VM to VM#8, thus if GPIO4 is re-triggered while the Voice Macro is running it will execute the power down voice macro rather than start the play sequence again.

The next command sets the LRMP bit of REG1, under normal operation the compressor ramps signal level to zero after a sound sample is played to prevent a DC voltage appearing on the output. The LRMP bit prevents this from happening while a sample is looping allowing continuous audio. To loop a sound sample, the audio should be edited such that the last sample loops smoothly to the first. To do this, create the sample in a sound editor at the sample rate desired for storage then find the first sample that returns to the initial condition and cut back audio to one before this sample. Note that tones require different lengths to fulfill these conditions at a given sample rate and thus loop numbers vary to produce the same length of output audio.

At the end of the VM REG1 is reset and the trigger is re-enabled back to VM#7 before powering down.

- **VM#7: R4_PlayLoop (GPIO4)**
  a. CFG(R4, 0x08) ; Configure GPIO4 to execute VM# 8 on next trigger.
  b. CFG(REG1, 0x20) ; Configure LRMP bit in REG1
  c. LOOP_VP(Do,20) ; LOOP “Do” 20 times.
  d. LOOP_VP(Re,250) ; LOOP “Re” 250 times.
  e. LOOP_VP(Mi,5) ; LOOP “Mi” 5 times.
  f. LOOP_VP(Fa,33) ; LOOP “Fa” 33 times.
  g. LOOP_VP(So,10) ; LOOP “So” 10 times
  h. LOOP_VP(La,10) ; LOOP “La” 10 times
  i. LOOP_VP(Si,7) ; LOOP “Si” 7 times.
  j. Silence (128 ms) ; Insert 128ms of silence
  k. CFG(REG1, 0x00) ; Reset REG1
  l. CFG(R4, 0x07) ; Configure GPIO4 to execute VM#7 on next trigger.
  m. PD ; Power Down
- **VM#8: PD_R4**
  a. CFG(REG1, 0x00) ; Configure Register one to its default value 00
  b. CFG(R4, 0x07) ; Configure GPIO4 to execute VM#7 on next trigger.
  c. PD ; Power Down

#### 10.1.4 Example: Uninterruptable Trigger, smooth audio.

In this example a single trigger on GPIO3 will sequence through several messages until all messages are played the playback cannot be interrupted by any other trigger. The example also demonstrates how to use begin and end segments to create smooth playback. Each "note" consists of concatenating three voice prompts, for instance "So_begin" "So" and "So_end". The begin and end prompts ramp the audio smoothly to avoid sudden transients in sound level. The middle, full amplitude, section is created by looping a short sample.

At the beginning of the Voice Macro, all triggers are disabled so that Voice Macro cannot be interrupted from any source. The NRMP bit of REG1 is set so that concatenation of audio occurs without any ramp down between prompts. At the end of the macro, interrupts are re-enabled and device is powered down.

- **VM#9: R3_Non-Int_Smooth (GPIO3)**
  a. CFG(REG_GPIO_AF1, 0x00) ; Disable all triggers.
  b. CFG(REG1, 0x04) ; Set NRMP bit
  c. PLAY_VP(So_begin) ; Play "So_begin"
  d. LOOP_VP(So,10) ; Loop "So" 10 times.
  e. PLAY_VP(So_end) ; Play "So_end"
  f. PLAY_VP(Fa_begin)
  g. LOOP_VP(Fa,33)
  h. PLAY_VP(Fa_end)
  i. PLAY_VP(Mi_begin)
  j. LOOP_VP(Mi,5)
  k. PLAY_VP(Mi_end)
  l. PLAY_VP(Re_begin)
  m. LOOP_VP(Re,250)
  n. PLAY_VP(Re_end)
  o. PLAY_VP(Do_begin)
  p. LOOP_VP(Do,20)
  q. PLAY_VP(Do_end)
  r. Wait Interrupt ; Wait for audio to finish
  s. CFG(REG1, 0x00) ; Reset NRMP bit
  t. CFG(REG_GPIO_AF1, 0x3f) ; Re-enable interrupts
  u. PD ; Power down device.

### 10.1.5 Example: Continuous Play until re-trigger.

In this example a single trigger on GPIO2 will sequence through several messages with pause in between each message. Messages are played in a loop indefinitely until another trigger occurs on GPIO2 to stop playback.

- **VM0#A: R2_Loop_VM (GPIO2)**
  a. CFG(R2, 0x0b) ; Set Trigger to VM#B (PD_R2)
  b. PLAY_VP(One) ; Play “One”
  c. Silence (256 ms) ; pause 256ms
  d. PLAY_VP(two) ; Play “Two”
  e. Silence (256 ms)
  f. PLAY_VP(three)
  g. Silence (736 ms)
  h. PLAY_VP(four)
  i. Silence (256 ms)
  j. EXE_VM(0xA) ; Execute VM#A (repeat)
  k. Finish
- **VM0#B: PD_R2**
  a. CFG(R2, 0x0a) ; Reset Trigger to VM#A
  b. PD ; Power Down.

### 10.1.6 Example: Level Hold Trigger.

In this example holding GPIO1 will play several messages. Releasing GPIO1 will stop the playback. No other triggers will affect operation.

- **VM#C: R1_Level_Hold (GPIO1)**
  a. CFG(REG_GPIO_AF0, 0x02) ; Enable rising edge trigger for GPIO2
  b. CFG(REG_GPIO_AF1, 0x02) ; Disable all triggers except GPIO2
  c. CFG(R1, 0x0d) ; Set Trigger to VM#D (PD_R1)
  d. CFG(REG1, 0x20)
  e. LOOP_VP(Re,200)
  f. Silence (32 ms)
  g. LOOP_VP(Mi,4)
  h. Silence (32 ms)
  i. LOOP_VP(Fa,20)
  j. Silence (32 ms)
  k. CFG(REG1, 0x00)
  l. PLAY_VP(applause)
  m. PD

- **VM#D: PD_R1**
    a. CFG(REG_GPIO_AF0, 0x00) ; Disable rising edge trigger
    b. CFG(REG_GPIO_AF1, 0x3f) ; Re-enable all triggers.
    c. CFG(REG1, 0x00) ; Ensure REG1 reset
    d. CFG(R1, 0x0c) ; Set trigger to VM#C
    e. PD ; Power Down.

## 11 INITIALIZATION FLOWCHART

Whenever the ISD2100 detects a power-on reset condition, or a SPI RESET command, it begins a power-on initialization (POI) sequence. Whenever the ISD2100 receives a power up command (PU) when it is in a power down state, it begins a power-up initialization (PU) sequence. Voice Macros VM#0 and VM#1 are reserved for POI and PU initialization routines. If the vectors VM#0 or VM#1 do not exist, then a default routine is executed. The default sequence for POI is to power-down the ISD2100. The default PU sequence is to select a clock configuration of internal oscillator.

GPIO-Wakeup Voice Macro has the same flow as PU Voice Macro. If a GPIO trigger wakes up the device from power down the device will execute VM#2 rather than the PU macro VM#1.

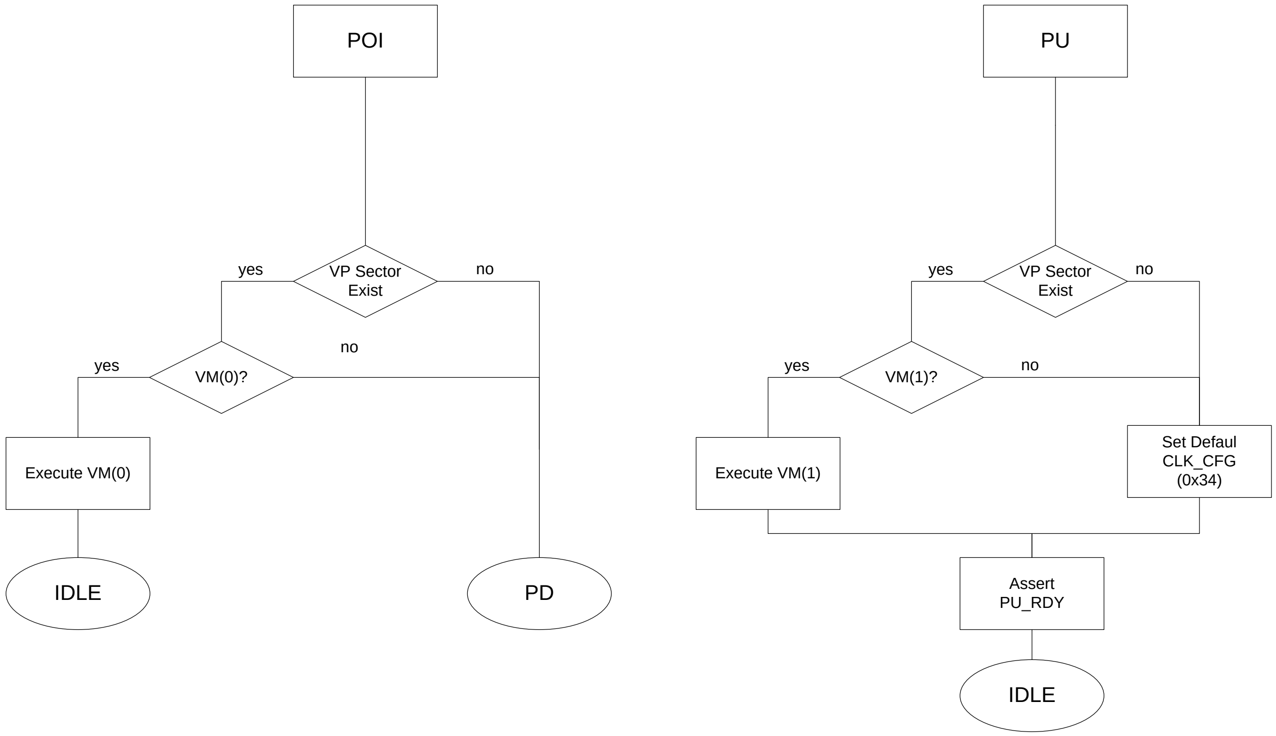



Figure 11-1 POI and PU Initialization Flowcharts

## 12 DEVICE CONFIGURATION AND STATUS

### 12.1 CLOCK CONFIGURATION

The clock configuration register is accessed via the SET_CLK_CFG command. It configures the clock source of the ISD2100. The default state of the clock configuration register is 0x00, in which the oscillator is referenced to the internal resistor.

Table 12-1 Clock Configuration Register Description

| CLK_CFG | | | | | | | |
|---|---|---|---|---|---|---|---|
| **Bit 7** | **Bit 6** | **Bit 5** | **Bit 4** | **Bit 3** | **Bit 2** | **Bit 1** | **Bit 0** |
| - | - | - | - | - | - | CLK_INP_SEL[1:0] | |

Default 0x00:

- Internal clock with internal reference

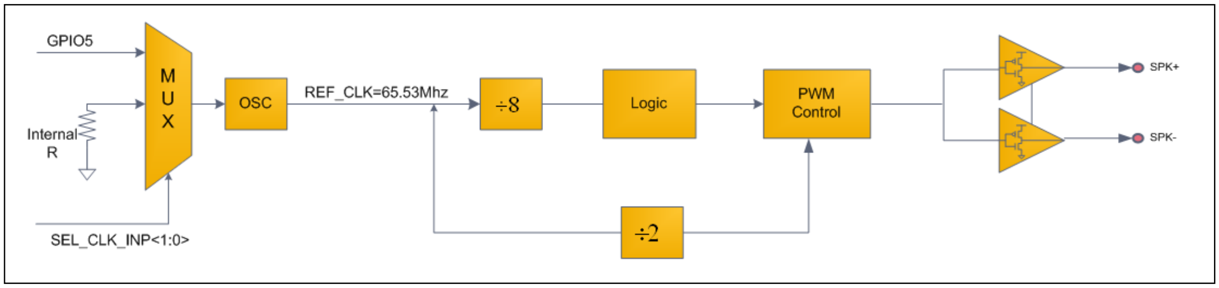



Figure 12-1 ISD2100 Clock Configuration

Table 12-2 Clock Configuration Source

| CLK_INP_SEL | |
|---|---|
| **CLK_INP_SEL[1:0]** | **Clock Source** |
| 00 | Internal Oscillator with Internal Resistor |

## 12.2 Device Status Register

Whenever the ISD2100 receives an SPI command it also returns its current status via MISO. The details of the status byte are shown below. For commands that are not reading digital data from the device this status byte is sent via MISO for every byte of data sent to the ISD2100.

Table 12-3 Status Register Description

| Status Byte | | | | | | | |
|---|---|---|---|---|---|---|---|
| **Bit 7** | **Bit 6** | **Bit 5** | **Bit 4** | **Bit 3** | **Bit 2** | **Bit 1** | **Bit 0** |
| PD | DBUF_RDY | INT | - | - | VM_BSY | CBUF_FUL | CMD_BSY |

The individual bits of the status register refer to the following conditions:

- **PD** – If this bit is high then the device is powered down. The DBUF_RDY bit will be low. When PD is high only the READ_STATUS, READ_INT and PWR_UP commands are accepted. If any other command is sent, it is ignored and no interrupt for an error is generated.
- **DBUF_RDY** – in PD this bit is low indicating the device can only accept a PWR_UP (power up) command. When PD is low this bit reflects the state of the RDY/BSY pin.
- **INT** – an interrupt has been generated. The interrupt is cleared by the READ_INT command. Interrupt type can be determined by the bits of the Interrupt Status Byte.
- **VM_BSY** – indicates the device is processing a voice macro. The device will not respond to a new audio command until this bit returns low.
- **CBUF_FUL** – indicates that the command buffer is full. No more commands can be queued for execution until this bit returns low.
- **CMD_BSY** – indicates the device is processing a command. Device will not respond to a new command until this bit returns low. If CMD_BSY=1 and CBUF_FUL=0 and VM_BSY=0, a new command will go into the command buffer and execute when the current command finishes. If CMD_BSY=1 and CBUF_FUL=1 or VM_BSY=1 any new audio command will be ignored and generate a command error. For erasing commands like ERASE_MEM and CHIP_ERASE, the user should poll this bit to determine if erase operation is complete.

Whenever the ISD2100 generates an interrupt the Interrupt Status register holds flags that indicate what type of interrupt was generated. These flags will remain set until a READ_INT command clears them and the interrupt pin.

Table 12-4 Interrupt Status Register Description

| Interrupt Status Byte | | | | | | | |
|---|---|---|---|---|---|---|---|
| **Bit 7** | **Bit 6** | **Bit 5** | **Bit 4** | **Bit 3** | **Bit 2** | **Bit 1** | **Bit 0** |
| - | MPT_ERR | WR_FIN | CMD_ERR | OVF_ERR | CMD_FIN | - | - |

The individual bits of the status register refer to the following conditions:

**INT** – an interrupt has been generated. The interrupt is cleared by the READ_INT command.

- **MPT_ERR** – Indicates a memory protection error. Digital access attempted for protected memory.
- **WR_FIN** – indicates a digital write command has finished writing to the flash memory.
- **CMD_ERR** – an invalid command was sent to the device. Command was ignored because the command buffer was full, a voice macro was active or the device was not ready to respond to an erase command.
- **OVF_ERR** – This error is generated if host illegally tries to read or write data while RDY/BSYB pin is low. It is also generated if a digital read or write attempts to read or write past the end of memory.
- **CMD_FIN** – This bit indicates an interrupt was generated because a command finished executing. A CMD_FIN interrupt will be generated each time an audio play or voice macro finishes.

## 12.3 DEVICE CONFIGURATION REGISTERS

This section describes the configuration registers of the ISD2100.

### 12.3.1 CFG_REG0 – Sample Rate Override

Register CFG_REG0 allows controls the sample rate during message playback operations. Sample rate is normally set automatically by the Voice Prompt, to override sample rate set bit 0 of CFG1 high, then SR[2:0] = CFG_REG0[7:5] will control the sample rate.

| Address | Access Mode | Value At Reset |
|---|---|---|
| 0x00 | R/W | 0x64 |

| Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|
| SR[2:0] | | | NA | | | | |

SR[2:0] Allow override of playback sample rate. Details in Table 12-5

Table 12-5 CFG0 Sample Rate Control

| Ratio to Fs | Sample Rate (kHz) (for Fs=32kHz) | Code SR[2:0] | CFG0[7:0] (Dec) | (Hex) |
|---|---|---|---|---|
| 8 | 4 | 0 | 0 | 0x00 |
| 6 | 5.333 | 1 | 32 | 0x20 |
| 5 | 6.4 | 2 | 64 | 0x40 |
| 4 | 8 | 3 | 96 | 0x60 |
| 2.5 | 12.8 | 4 | 128 | 0x80 |
| 2 | 16 | 5 | 160 | 0xA0 |
| 1 | 32 | 6 | 192 | 0xC0 |

The current operational mode of the ISD2100 can also be queried by setting CFG_REG1[4] and reading CFG_REG0. Under this condition, rather than reading back the configuration register, the result will be current audio path sample rate.

### 12.3.2 CFG_REG1 – Decompression Control

Configuration register CFG_REG1 controls how compressed audio is treated by the compression block.

| Address | Access Mode | Value At Reset |
|---|---|---|
| 0x01 | R/W | 0x00 |

| Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|
| Reserved | Reserved | LRMP | CFG0_READ | Reserved | NRMP | SRSIL | SRCFG |

Configuration register CFG1 controls how compressed audio is treated by the compression block:

SRCFG — Forces the sample rate to be set by the CFG_REG0 register. By setting this bit to one sample rate contained in the stored audio header is overridden.

SRSIL — Under normal circumstances, whenever a change in sample rate is detected between two consecutive messages a period of silence is automatically inserted. This is to prevent any transients in the signal path occurring as filter coefficients are changed. To turn off this silence insertion set this bit to one.

NRMP — Under normal conditions, if an audio playback finishes at a non-zero level the input to the signal path will be ramped to zero. This prevents a DC offset appearing on the output. To turn this feature off, for instance if a small audio sample is being

| | |
|---|---|
| | looped, set this bit to one. Please note that **NRMP** and **LRMP** should not be set at the same time. |
| CFG0_READ | When this bit is set, a read of CFG_REG0 will read the current sample rate rather than the setting of the CFG_REG0 register. |
| LRMP | Set this bit to 1, the input to the signal path will not be ramped during the loop-play, but will be ramped to zero when the playback finishes or being stopped by a STOP or STOP_LP command. Please note that **NRMP** and **LRMP** should not be set at the same time. |

Default 0x00:

- Sample rate is set by the audio header.
- Whenever a change in sample rate is detected between two consecutive messages, a period of silence is automatically inserted.
- If an audio playback finishes at a non-zero level the input to the signal path will be ramped to zero.
- A read of CFG_REG0 will read the setting of the CFG_REG0 register rather than the current sample rate.

### 12.3.3 CFG_REG2 – Compression Source

Configuration register CFG_REG2 controls how digital audio signal path is configured.

| Address | Access Mode | Value At Reset |
|---|---|---|
| 0x02 | R/W | 0x00 |

| Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|
| Reserved | DECODE | SPI_IN | Reserved | Reserved | PWM_OUT | Reserved | SPI_OUT |

| | |
|---|---|
| SPI_OUT | Output signal data to SPI. If DECODE is selected, the signal path is bypassed and ISD2100 is ready for SPI Playback operation. Use the SPI_PCM_READ command to read the audio data out the SPI interface. |
| SPI_IN | Input data from SPI. |
| PWM_OUT | Output signal to PWM speaker driver. If PWM_OUT=0 speaker pins are tri-state. If PWM_OUT=1 speaker pins will be driven and PWM modulator active. |
| DECODE | Enable the De-compression block. |

Default 0x00:

- All digital paths are disabled.

### 12.3.4 VOLC – Volume Control

| Address | Access Mode | Value At Reset | | | | | |
|---|---|---|---|---|---|---|---|
| 0x03 | R/W | 0x00 | | | | | |
| Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
| VOLC[7:0] | | | | | | | |

| | |
|---|---|
| VOLC[7:0] | Sets the output signal volume. Setting 0 has 0dB attenuation. Each subsequent step provides 0.25dB of attenuation. |

### 12.3.5 CFG_REG4 – Checksum Reset

| Address | Access Mode | Value At Reset | | | | | |
|---|---|---|---|---|---|---|---|
| 0x04 | R/W | 0x00 | | | | | |
| Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
| Reserved | Reserved | Reserved | RST_CHKSUM | Reserved | Reserved | Reserved | Reserved |

| | |
|---|---|
| RST_CHKSUM | Write a 1 followed by a 0 to this bit to reset the CHECKSUM calculation. |

### 12.3.6 CFG_REG9 – PWM Control

| Address | Access Mode | Value At Reset | | | | | |
|---|---|---|---|---|---|---|---|
| 0x09 | R/W | 0x00 | | | | | |
| Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
| PWM_FREQ | | | EN_DITHER | | Reserved | Reserved | Reserved |

| | |
|---|---|
| PWM_FREQ | Sets nominal PWM carrier frequency according to Table 12-6. |
| EN_DITHER | Spread the PWM carrier frequency by dithering. EN_DITHER[1:0] = 3 is the highest dithering option whereas EN_DITHER[1:0] = 0 means no dithering. |

Table 12-6 PWM Carrier Frequency

| PWM_FREQ[2:0] | Nominal carrier frequency |
|---|---|
| 000 | 287kHz |
| 001 | 420kHz |
| 010 | 862kHz |
| 011 | 1.26MHz |
| 100 | 84kHz |
| 101 | 125kHz |
| 110 | 166kHz |
| 111 | 245kHz |

### 12.3.7 Checksum Registers

The SPI command CHECKSUM initiates a 4-byte checksum calculation from beginning of memory to a specified end address. To start a checksum calculation, user should first reset the circuit by writing one to register CFG_REG4 bit-4 followed by a zero, and then issue the CHECKSUM SPI command to start the calculation. The calculation is based on Fletcher-32 algorithm, and the calculated checksum is stored in registers 0x10 ~ 0x13, with the order of CHK_SUM1[7:0], CHK_SUM1[15:8], CHK_SUM2[7:0] and CHK_SUM2[15:8] respectively.

| Address | Access Mode | Value At Reset | Nominal Value |
|---|---|---|---|
| 0x10-0x13 | R | 0x0000 | |

| | Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 |
|---|---|---|---|---|---|---|---|---|
| 0x10 | CHK_SUM1[7:0] | | | | | | | |
| 0x11 | CHK_SUM1[15:8] | | | | | | | |
| 0x12 | CHK_SUM2[7:0] | | | | | | | |
| 0x13 | CHK_SUM2[15:8] | | | | | | | |

### 12.3.8 GPIO_DO – GPIO Data Out

| Address | Access Mode | Value At Reset | Nominal Value |
|---|---|---|---|
| 0x19 | R/W | 0x00 | |

| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 |
|---|---|---|---|---|---|---|---|
| - | - | GPIO_DO | | | | | |

| | |
|---|---|
| GPIO_DO | This register contains the data to be written out the GPIO pins GPIO[5:0]. Data will propagate to the pin if the corresponding GPIO_OE bit is enabled. |

### 12.3.9 GPIO_OE – GPIO Output Enable

| Address | Access Mode | Value At Reset | Nominal Value |
|---|---|---|---|
| 0x1A | R/W | 0x00 | |

| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 |
|---|---|---|---|---|---|---|---|
| - | - | GPIO_OE | | | | | |

GPIO_OE — This register contains output enable mask for the GPIO pins GPIO[5:0]. If a GPIO_OE bit is high, data in the GPIO_DO register will propagate to the pin. If low the driver of the pin is in a tri-state condition. GPIO_OE[1]=1 has no effect as the output on this pin is disabled to prevent conflict with SPI functionality.

### 12.3.10 GPIO_PE – GPIO Pull Enable

| Address | Access Mode | Value At Reset | Nominal Value |
|---|---|---|---|
| 0x1B | R/W | 0xFF | |

| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 |
|---|---|---|---|---|---|---|---|
| - | - | GPIO_PE | | | | | |

GPIO_PE — This register contains the pull enable mask for the GPIO pins GPIO[5:0]. If a GPIO_PE bit is high then a pull-up or pull-down is connected to the pin depending on the state of GPIO_PS. Default value is to enable pull-ups on all GPIO pins.

### 12.3.11 GPIO_DIN – GPIO Data In

| Address | Access Mode | Value At Reset | Nominal Value |
|---|---|---|---|
| 0x1C | R | 0xXX | |

| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 |
|---|---|---|---|---|---|---|---|
| - | - | GPIO_DIN | | | | | |

GPIO_DIN — This read only register reflects the state of the GPIO pins GPIO[5:0].

### 12.3.12 GPIO_PS – GPIO Pull Select

| Address | Access Mode | Value At Reset | Nominal Value |
|---|---|---|---|
| 0x1D | R/W | 0xFF | |

| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 |
|---|---|---|---|---|---|---|---|
| - | - | GPIO_PS | | | | | |

| | |
|---|---|
| GPIO_PS | This register contains the pull select mask for the GPIO pins GPIO[5:0]. If a GPIO_PS bit is high then a pull-up is selected, if low a pull-down is selected. Selection only propagates to pin when corresponding GPIO_PE bit is also set. Default is to select pull-up. |

### 12.3.13 GPIO_AF – GPIO Alternate Function Control

| Address | Access Mode | Value At Reset | Nominal Value |
|---|---|---|---|
| 0x1E-0x1F | R/W | 0x0000 | |

| | Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 |
|---|---|---|---|---|---|---|---|---|
| 1E | - | - | GPIO_AF1 | | | | | |
| 1F | - | - | GPIO_AF0 | | | | | |

These two registers control the functionality of the GPIO pins GPIO[5:0]. The default configuration is to configure all pins as GPIO pins. GPIO[2:0] have alternate functions as SPI interface pins. Activating the SPI interface by pulling SSB low will automatically select alternate function mode for GPIO[2:0].If SPI operations requiring RDY/BSY handshaking are required, then GPIO[4] must be set to alternate function mode. Section 6.3 explains the action of these registers in detail.

### 12.3.14 R0-R7 – Indirect Jump Registers

R0-R7 are 16 bit indirect jump registers used to store jump indexes for GPIO trigger events and the indirect SPI commands PLAY_VP@Rn PLAY_VP_LP@Rn and EXE_VM@Rn. R0 is stored in register addresses 0x20 and 0x21 LSB first. The table below shows the address locations of R0-R7.

| Address | Access Mode | Value At Reset | Nominal Value |
|---|---|---|---|
| 0x20-0x2F | R/W | 0x0000 | |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Addr | 0x20 | 0x21 | 0x22 | 0x23 | 0x24 | 0x25 | 0x26 | 0x27 |
| Reg | R0[7:0] | R0[15:8] | R1[7:0] | R1[15:8] | R2[7:0] | R2[15:8] | R3[7:0] | R3[15:8] |
| Addr | 0x28 | 0x29 | 0x2A | 0x2B | 0x2C | 0x2D | 0x2E | 0x2F |
| Reg | R4[7:0] | R4[15:8] | R5[7:0] | R5[15:8] | R6[7:0] | R6[15:8] | R7[7:0] | R7[15:8] |

When a GPIO pin is configured as a trigger, the GPIO pin is associated with the corresponding jump register. That is GPIO0 with R0, GPIO1 with R1 etc.. If GPIO0 is triggered, the ISD2100 will execute the Voice Macro whose index is stored in R0. Thus if R0 is initialized to 0x0012, then Voice Macro #12 will be executed upon a trigger. The same effect would occur if a EXE_VM@R0 SPI command was sent.

### 12.4 Device Identification Registers.

By sending the command READ_ID the device responds with four-byte identification. The first byte reports the ISD2100 family version. The following three bytes are a JEDEC compliant code indicating the memory type. The first byte is the manufacturer code which is 0xEF. The next byte is for memory type which is 0x20. The last byte is memory size according to the following table:

Table 12-7 Memory Size ID Byte

| ISD Part Number | Capacity | SIZE_ID | End Address |
|---|---|---|---|
| ISD2130 | 1Mb | 0x11 | 0x1FFFF |
| ISD2115A | 64Kbyte | 0x10 | 0x0FFFF |

## 13 SPI COMMANDS

The ISD2100 provides SPI commands to play audio, query device status, perform digital memory operations and configure the device. The following section contains a list of all SPI commands and their describes their function.

Table 13-1 SPI Commands

| Instructions | Byte 0 | Byte 1 | Byte 2 | Byte 3 | Byte 4 … Byte n | Description |
|---|---|---|---|---|---|---|
| **PLAY_VP** | 0xA6 | INX[15:8] | INX[7:0] | | | Play Voice Prompt Index INX |
| **PLAY_VP@Rn** | 0xAE | *n* = 0 …7 | | | | Play Voice Prompt; Index is value in register R*n*. |
| **PLAY_VP_LP** | 0xA4 | INX[15:8] | INX[7:0] | CNT[15:8] | CNT[7:0] | Loop Play Voice Prompt Index INX, CNT times. |
| **PLAY_VP_LP@ Rn** | 0xB2 | *n* = 0 …7 | CNT[15:8] | CNT[7:0] | | Loop Play Voice Prompt; Index in register R*n*, CNT times. |
| **STOP_LP** | 0x2E | | | | | Stop Loop Play Voice Prompt |
| **EXE_VM** | 0xB0 | Index[15:8] | Index[7:0] | | | Execute voice macro Index |
| **EXE_VM@Rn** | 0xBC | *n* = 0 …7 | | | | Execute voice macro; Index contained in register R*n* |
| **PLAY_SIL** | 0xA8 | LEN[7:0] | | | | Play silence for LEN*32ms |
| **STOP** | 0x2A | | | | | STOP current playback operation. |
| **SPI_PCM_READ** | 0xAC | D0[7:0] | D0[15:8] | D1[7:0] | D1[15:8] …Dn[7:0] Dn[15:8] | Receive 16 bit PCM audio data [low-byte, high-byte] from ISD2100 via SPI interface. |
| **SPI_SND_DEC** | 0xC0 | D0[7:0] | D1[7:0] | D2[7:0] | D3[7:0] …Dn[7:0] | Send compressed audio data to ISD2100 via SPI interface for decoding. |

| Instructions | Byte 0 | Byte 1 | Byte 2 | Byte 3 | Byte 4 … Byte n | Description |
|---|---|---|---|---|---|---|
| **READ_STATUS** | 0x40 | XX | XX | XX | … | Query status of ISD2100. |
| **READ_INT** | 0x46 | XX | XX | XX | … | Query status and clear interrupt flags of ISD2100. |
| **READ_ID** | 0x48 | XX | XX | XX | XX | Read device ID of ISD2100. |
| **DIG_READ** | 0xA2 | A[23:16] | A[15:8] | A[7:0] | XX, … XX | Read digital data from address A. |
| **DIG_WRITE** | 0xA0 | A[23:16] | A[15:8] | A[7:0], | D0[7:0], … Dn[7:0] | Write digital data from address A. |
| **ERASE_MEM** | 0x24 | SA[23:16] | SA[15:8] | SA[7:0] | | Erase 1kByte sector of memory containing start address SA. |
| **CHIP_ERASE** | 0x26 | 0x01 | | | | Initiate a mass erase of memory. |
| **CHECKSUM** | 0xF2 | EA[23:16] | EA[15:8] | EA[7:0] | | Calculate checksum from 0x0000 to the specified end address EA. |
| **PWR_UP** | 0x10 | | | | | Power up ISD2100 |
| **PWR_DN** | 0x12 | | | | | Power down ISD2100 |
| **SET_CLK_CFG** | 0xB4 | CFG_CLK[7:0] | | | | Set clock configuration register. |
| **RD_CLK_CFG** | 0xB6 | XX | | | | Read clock configuration register. |
| **WR_CFG_REG** | 0xB8 | REG[7:0] | D0[7:0], …Dn[7:0] | | | Write data D0…Dn to configuration register(s) starting at configuration register REG. |
| **RD_CFG_REG** | 0xBA | REG[7:0] | XX, …XX | | | Read configuration register(s) starting at configuration register REG. |
| **RESET** | 0x14 | | | | | Resets all the registers and initiates POI procedure. |

Each command will be accepted if certain conditions are met as in the following table, or a CMD_ERR interrupt will be generated and the command ignored. It is good practice to check device status prior to sending commands to ensure device is ready to accept the command.

Table 13-2 Commands vs. Status

| | | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|---|
| **Instructions** | **Op Code** | **PD** | **DBUF_RDY** | **INT** | **-** | **-** | **VM_BSY** | **CBUF_FUL** | **CMD_BSY** |
| **PLAY_VP** | 0xA6 | 0 | 1 | x | - | - | 0 | 0 | x |
| **PLAY_VP@Rn** | 0xAE | 0 | 1 | x | - | - | 0 | 0 | x |
| **PLAY_VP_LP** | 0xA4 | 0 | 1 | x | - | - | 0 | 0 | x |
| **PLAY_VP_LP@Rn** | 0xB2 | 0 | 1 | x | - | - | 0 | 0 | x |
| **STOP_LP** | 0x2E | 0 | 1 | x | - | - | x | x | x |
| **EXE_VM** | 0xB0 | 0 | 1 | x | - | - | 0 | 0 | 0 |
| **EXE_VM@Rn** | 0xBC | 0 | 1 | x | - | - | 0 | 0 | 0 |
| **PLAY_SIL** | 0xA8 | 0 | 1 | x | - | - | 0 | 0 | x |
| **STOP** | 0x2A | 0 | 1 | x | - | - | x | x | x |
| **SPI_PCM_READ** | 0xAC | 0 | 1 | x | - | - | x | x | x |
| **SPI_SND_DEC** | 0xC0 | 0 | 1 | x | - | - | 0 | 0 | 0 |
| **READ_STATUS** | 0x40 | x | x | x | - | - | x | x | x |
| **READ_INT** | 0x46 | x | x | x | - | - | x | x | x |
| **READ_ID** | 0x48 | 0 | 1 | x | - | - | x | x | x |
| **DIG_READ** | 0xA2 | 0 | 1 | x | - | - | 0 | 0 | 0 |
| **DIG_WRITE** | 0xA0 | 0 | 1 | x | - | - | 0 | 0 | 0 |
| **ERASE_MEM** | 0x24 | 0 | 1 | x | - | - | 0 | 0 | 0 |
| **CHIP_ERASE** | 0x26 | 0 | 1 | x | - | - | 0 | 0 | 0 |
| **CHECKSUM** | 0xF2 | 0 | 1 | x | - | - | 0 | 0 | 0 |
| **PWR_UP** | 0x10 | 1 | 0 | x | - | - | x | x | x |
| **PWR_DN** | 0x12 | 0 | 1 | x | - | - | x | x | x |
| **SET_CLK_CFG** | 0xB4 | 0 | 1 | x | - | - | 0 | 0 | 0 |
| **RD_CLK_CFG** | 0xB6 | 0 | 1 | x | - | - | x | x | x |
| **WR_CFG_REG** | 0xB8 | 0 | 1 | x | - | - | x | x | x |
| **RD_CFG_REG** | 0xBA | 0 | 1 | x | - | - | x | x | x |
| **RESET** | 0x14 | x | x | x | - | - | x | x | x |

### 13.1 Audio Play Commands

This section describes the 11 audio commands that can be sent to the device.

#### 13.1.1 PLAY_VP – Play Voice Prompt

| **PLAY_VP** | | | | |
|---|---|---|---|---|
| Byte Sequence: | Host controller | 0xA6 | *Index*[15:8] | *Index*[7:0] |
| | ISD2100 | Status Byte | Status Byte | Status Byte |
| Description: | Play Voice Prompt *Index* | | | |
| Interrupt Generation: | CMD_ERR if not accepted. CMD_FIN when playback is finished. | | | |

This command initiates a play of a pre-recorded voice-prompt. Before execution of command a valid signal path must be set up, and the device must have space in the audio command buffer. After completion of playback, the device will generate an interrupt. The command will be accepted if status bits PD=0, DBUF_RDY=1, VM_BSY=0 and CBUF_FUL=0. If any of these conditions are not met then a CMD_ERR interrupt will be generated and the command ignored. If command is terminated after the command byte is sent no interrupt will be generated. Once playback is finished a CMD_FIN interrupt will be generated.

#### 13.1.2 PLAY_VP@Rn – Play Voice Prompt @Rn, n = 0 ~ 7

| **PLAY_VP@Rn** | | | | |
|---|---|---|---|---|
| Byte Sequence: | Host controller | 0xAE | *n = 0 ~ 7* | |
| | ISD2100 | Status Byte | Status Byte | |
| Description: | Play Voice Prompt, *Index@Rn* | | | |
| Interrupt Generation: | CMD_ERR if not accepted. CMD_FIN when playback is finished. | | | |

This command function is the same as PLAY_VP except that the 16bit index is stored in jump register R*n*, *n* = 0 ~ 7.

### 13.1.3 PLAY_VP_LP – Play Voice Prompt, Loop

| **PLAY_VP_LP** | | | | | | |
|---|---|---|---|---|---|---|
| Byte Sequence: | Host controller | 0xA4 | *Index*[15:8] | *Index*[7:0] | *LoopCnt*[15:8] | *LoopCnt*[7:0] |
| | ISD2100 | Status Byte | Status Byte | Status Byte | Status Byte | Status Byte |
| Description: | Play Voice Prompt *Index, LoopCnt* times. | | | | | |
| Interrupt Generation: | CMD_ERR if not accepted. CMD_FIN when playback is finished. | | | | | |

This command initiates a loop-play of a pre-recorded voice-prompt. Number of play-loops is specified in *LoopCnt*[15:0]. Setting *LoopCnt* to 0 initiates an endless play, which can be ended by a STOP or STOP_LP command. Before execution of command a valid signal path must be set up, and the device must have space in the audio command buffer. After completion of playback, the device will generate an interrupt. The command will be accepted if status bits PD=0, DBUF_RDY=1, VM_BSY=0 and CBUF_FUL=0. If any of these conditions are not met, then a CMD_ERR interrupt will be generated and the command ignored. If command is terminated after the command byte is sent no interrupt will be generated. Once playback is finished a CMD_FIN interrupt will be generated.

### 13.1.4 PLAY_VP_LP@Rn – Play Voice Prompt @Rn, Loop

| **PLAY_VP_LP@Rn** | | | | | |
|---|---|---|---|---|---|
| Byte Sequence: | Host controller | 0xB2 | *n = 0 ~ 7* | LoopCnt[15:8] | LoopCnt[7:0] |
| | ISD2100 | Status Byte | Status Byte | Status Byte | Status Byte |
| Description: | Play Voice Prompt, Loop, *Index@Rn* | | | | |
| Interrupt Generation: | CMD_ERR if not accepted. CMD_FIN when playback is finished. | | | | |

This command is same as PLAY_VP_LP except that the 16bit index is stored in Rn, n = 0 ~ 7.

### 13.1.5 STOP_LP – Stop Loop-Play Command

| **STOP_LP** | | | | |
|---|---|---|---|---|
| Byte Sequence: | Host controller | 0x2E | | |
| | ISD2100 | Status Byte | | |
| Description: | Stop current loop-play command and flush command buffer. | | | |
| Interrupt Generation: | Command itself does not generate interrupt, only those commands that it is stopping. | | | |

This command stops any current PLAY_VP_LP or PLAY_VP_LP@Rn command active in the ISD2100. The STOP_LP command does not flush the audio command buffer; that is, any command queued in the buffer when a STOP_LP is issued will be executed thereafter. When device has finished the active command a CMD_FIN interrupt will be generated. If there is no active PLAY_VP_LP or PLAY_VP_LOOP@Rn command, then STOP will have no effect.

### 13.1.6 EXE_VM – Execute Voice Macro

| **EXE_VM** | | | | |
|---|---|---|---|---|
| Byte Sequence: | Host controller | 0xB0 | *Index*[15:8] | *Index*[7:0] |
| | ISD2100 | Status Byte | Status Byte | Status Byte |
| Description: | Play voice macro *Index* | | | |
| Interrupt Generation: | CMD_ERR if not accepted. CMD_FIN when playback is finished. | | | |

This command initiates the execution of a pre-recorded voice macro. After completion of the voice macro the device will generate a CMD_FIN interrupt. The command will be accepted if status bits PD=0, DBUF_RDY=1, VM_BSY=0, CBUF_FUL=0 and CMD_BSY=0. If any of these conditions are not met then a CMD_ERR interrupt will be generated and the command ignored. If command is terminated after the command byte is sent no interrupt will be generated. Once voice macro execution is finished a CMD_FIN interrupt will be generated.

### 13.1.7 EXE_VM@Rn – Execute Voice Macro @Rn, n = 0 ~ 7

| **EXE_VM@Rn** | | | | |
|---|---|---|---|---|
| Byte Sequence: | Host controller | 0xBC | *n = 0 ~ 7* | |
| | ISD2100 | Status Byte | Status Byte | |
| Description: | Play voice macro *Index@Rn* | | | |
| Interrupt Generation: | CMD_ERR if not accepted. CMD_FIN when playback is finished. | | | |

This command is same as EXE_VM except that the 16bit index is stored in Rn, n = 0 ~ 7.

### 13.1.8 PLAY_SIL – Play Silence

| **PLAY_SIL** | | | | |
|---|---|---|---|---|
| Byte Sequence: | Host controller | 0xA8 | *LEN*[7:0] | |
| | ISD2100 | Status Byte 0 | Status Byte 0 | |
| Description: | Play silence for *LEN**32ms | | | |
| Interrupt Generation: | CMD_ERR if not accepted. CMD_FIN when silence playback complete. | | | |

This command plays a period of silence to the signal path. Before execution of command a valid signal path must be set up and the device must have space in the audio command buffer. After completion, the device will generate an interrupt. The length of silence played is determined by the data byte, *LEN*, sent. Silence is played in 32ms increments, total silence played is *LEN**32ms.

The command will be accepted if status bits PD=0, DBUF_RDY=1, VM_BSY=0 and CBUF_FUL=0. If any of these conditions are not met then a CMD_ERR interrupt will be generated and the command ignored. If command is terminated after the command byte is sent no interrupt will be generated. Once silence play is finished a CMD_FIN interrupt will be generated.

### 13.1.9 STOP – Stop Command

| **STOP** | | | | |
|---|---|---|---|---|
| Byte Sequence: | Host controller | 0x2A | | |
| | ISD2100 | Status Byte | | |
| Description: | Stop current audio command and flush command buffer. | | | |
| Interrupt Generation: | Command itself does not generate interrupt, only those commands that it is stopping. | | | |

This command stops any current audio command active in the ISD2100. If a PLAY_VP, EXE_VM or PLAY_SIL command is active, playback is stopped immediately. The STOP command flushes the audio command buffer, that is any command queued in the buffer when a STOP is issued will not be executed. When device has finished the active command a CMD_FIN interrupt will be generated. STOP will not stop an ERASE_MEM operation. If there is no active command then STOP will have no effect.

### 13.1.10 SPI_PCM_READ – Read PCM Data from SPI

| **SPI_PCM_READ** | | | | | | | |
|---|---|---|---|---|---|---|---|
| Byte Sequence: | Host controller | 0xAC | | | | | |
| | ISD2100 | Status Byte | D0[7:0] | D0[15:8] | …. | Dn[7:0] | Dn[15:8] |
| Description: | Read audio data via SPI interface. | | | | | | |
| Interrupt Generation: | OVF_ERR if RDY/BSY violated. | | | | | | |

This command allows the user to receive audio data, in 16bit PCM format, from the SPI interface. Before execution of command a valid signal path must be set up.

When receiving audio data from memory (SPI playback), then: (1) signal path must be set up for SPI output from the compressor. (2) A valid play command is then sent; valid play commands include PLAY_VP, PLAY_VP@Rn, PLAY_VP_LP, PLAY_VP_LP@Rn, EXE_VM and EXE_VM@Rn. (3) Followed by the SPI_PCM_READ command. Multiple SPI_PCM_READ commands can be sent. (4) To finish receiving data a STOP command is sent and device will generate a CMD_FIN interrupt.

When the end of message is reached a CMD_FIN interrupt will be generated and zero will be sent as data. If the valid play command in step (2) is EXE_VM, then a CMD_FIN interrupt will be generated at the end of voice macro.

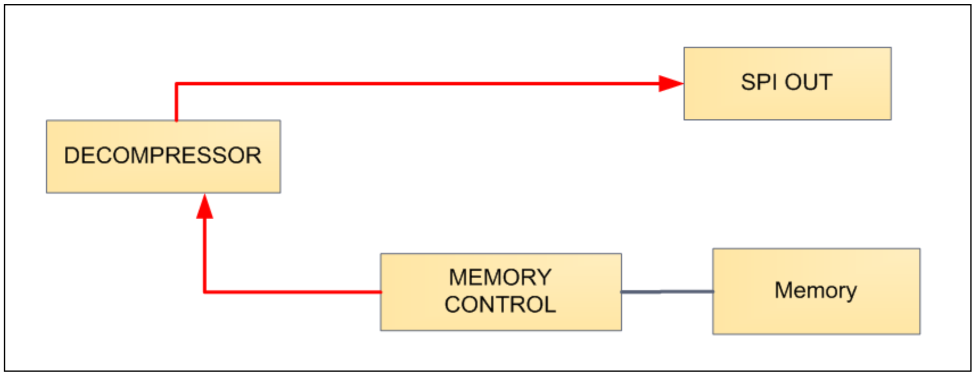



Figure 13-1 SPI Playback

The RDY/BSYB pin will go low whenever the internal FIFO is empty. If no path or playback operation is set up then RDY/BSYB pin will be low until command is terminated. If RDY/BSYB is ignored then an OVF_ERR interrupt is generated.

### 13.1.11 SPI Send Compressed Data to Decode

| **SPI_SND_DEC** | | | | | | |
|---|---|---|---|---|---|---|
| Byte Sequence: | Host controller | 0xC0 | D0[7:0] | D1[7:0] | …. | Dn[7:0] |
| | ISD2100 | Status Byte | | | | |
| Description: | Write compressed audio data via SPI interface. | | | | | |
| Interrupt Generation: | OVF_ERR if RDY/BSYB violated. | | | | | |

This command allows the user to send compressed audio data, in a byte formatted bit stream, down the SPI interface to the de-compressor and signal path.

Before execution of command

(1) A valid signal path must be set up. Valid paths are similar to a standard playback.
(2) Multiple SPI_SND_DEC commands can be issued to send data to the ISD 2100.
(3) To finish decoding a STOP command is sent and device will respond with a CMD_FIN interrupt. RDY/BSYB pin will handshake dataflow if device cannot accept any further data for decompression. If host cannot keep up with data rate audio output will be corrupted.

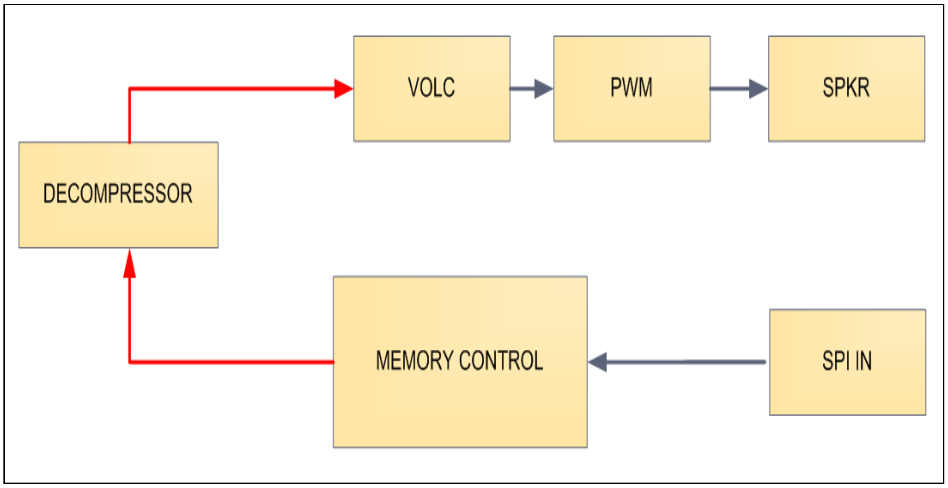



Figure 13-2 SPI Send Compressed Data to Decode

The RDY/BSYB pin will go low whenever the internal FIFO is full. If no path set up to accept audio data then RDY/BSYB pin will not return high until command is terminated. If RDY/BSYB is ignored then an OVF_ERR interrupt is generated. The SPI_SND_DEC command is accepted if no current play operation is active. If command is not accepted a CMD_ERR interrupt will be generated. It is possible to perform digital memory operations between SPI_SND_DEC operations, however care must be taken to maintain the required data rate to avoid audio corruption.

## 13.2 DEVICE STATUS COMMANDS.

### 13.2.1 Read Status

Powered up:

| **READ_STATUS** | | | |
|---|---|---|---|
| Byte Sequence: | Host controller | 0x40 | 0xXX |
| | ISD2100 | Status Byte | Interrupt Status Byte |
| Description: | Query device status. | | |

Powered down:

| **READ_STATUS** | | | |
|---|---|---|---|
| Byte Sequence: | Host controller | 0x40 | 0xXX |
| | ISD2100 | Status Byte 80h | 00h |
| Description: | Query device status. | | |

This command queries the ISD2100 device status. For details of device status see Section 12.2. If device is powered up, the two status bytes will be repeated for each two dummy bytes sent to the SPI interface. If device is powered down, only one status byte 80h shows up to the SPI interface at the same time the command is sent. This command is always accepted.

### 13.2.2 Read Interrupt

| **READ_INT** | | | | |
|---|---|---|---|---|
| Byte Sequence: | Host controller | 0x46 | 0xXX | |
| | ISD2100 | Status Byte | Interrupt Status Byte | |
| Description: | Query device status and clear interrupt flags. | | | |

This command queries the ISD2100 device status and clears any pending interrupts. After this command the hardware interrupt line will return inactive. The INT bit of the status register along with any status error bits will return inactive.

This command is accepted whenever device is powered up.

### 13.2.3 READ_ID - Read ISD2100 Device ID

| READ_ID | | | | | | |
|---|---|---|---|---|---|---|
| Byte Sequence: | Host controller | 0x48 | 0xXX | 0xXX | 0xXX | 0xXX |
| | ISD2100 | Status Byte | PART_ID | MAN_ID | MEM_TYPE | DEV_ID |
| Description: | Return ID of ISD2100 | | | | | |

This command queries the ISD2100 to returns four bytes to indicate the ISD2100 family member and the manufacturer, size and type of internal memory of the device.

The four bytes returned are:

- One byte ISD2100 Family ID, which is 0x03.
- Three bytes Flash JEDEC ID.
    1. MAN_ID=0xEF
    2. MEM_TYPE=0x20
    3. SIZE_ID (see Table 12-7)

## 13.3 DIGITAL MEMORY COMMANDS.

This section describes the digital data commands that can be sent to the device. Digital commands are ones that read, write or erase data in the flash memory. Digital and Audio commands cannot occur concurrently. To prevent possibility of data corruption, user should disable any Voice Macro triggers and check status to ensure device is idle (CMD_BSY=0) before sending any digital memory commands.

### 13.3.1 Digital Read

| DIG_READ | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Byte Sequence: | Host controller | 0xA2 | A[23:16] | A[15:8] | A[7:0] | 0xXX | … | 0xXX |
| | ISD2100 | Status | Status | Status | Status | D0 | … | Dn |
| Description: | Initiates a digital read of memory from address A [23:0]. | | | | | | | |
| Interrupt Generation: | ADDR_ERR if memory protected or RDY/BSYB violated. OVF_ERR if read past end of array. | | | | | | | |

This command initiates a read of flash memory from address A[23:0]. Following the three address bytes, data can be read out of memory in a sequential manner. The RDY/BSYB pin is used to control flow of data. If RDY/BSYB pin goes low, transfer must be paused until RDY/BSYB pin returns high. The user should check RDY/BSYB pin before every byte is sent/read including the command and address bytes. As many bytes of data as required can be read, command is terminated by raising SSB high, finishing the SPI transaction. If an attempt is made to read past the end of memory, status byte will be read back.

The command will always be accepted, and RDY/BSYB pin will go low until any active digital memory command is complete. If a digital read is attempted in read protected memory, status byte will be read back and an ADDR_ERR interrupt will be generated. If a read past the end of memory is attempted an OVF_ERR interrupt will be generated. If RDY/BSYB is violated then zero data will be read back and an **OVF_ERR** interrupt will be generated.

Digital and Audio commands cannot occur concurrently. To prevent possibility of data corruption, user should disable any Voice Macro triggers and check status to ensure device is idle (CMD_BSY=0) before sending a digital read command.

### 13.3.2 Digital Write

| **DIG_WRITE** | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Byte Sequence: | Host controller | 0xA0 | A[23:16] | A[15:8] | A[7:0] | D0 | … | Dn |
| | ISD2100 | Status | Status | Status | Status | Status | … | Status |
| Description: | Initiates a digital write to memory from address A[23:0]. | | | | | | | |
| Interrupt Generation: | ADDR_ERR if memory protected or RDY/BSYB violated. OVF_ERR if write past end of array. | | | | | | | |

This command initiates a write to flash memory from address A [23:0]. Following the three address bytes, data can be written to memory in a sequential manner. The RDY/BSYB pin is used to control flow of data. If RDY/BSYB pin goes low, transfer must be paused until RDY/BSYB pin returns high. The user should check RDY/BSYB pin before every byte is sent including the command and address bytes. As many bytes of data as required can be written, command is terminated by raising SSB high, finishing the SPI transaction.

The command will always be accepted, and RDY/BSYB pin will go low until any active digital memory command is complete. If a digital write is attempted in write protected memory, data will be ignored and an ADDR_ERR interrupt will be generated. If a write is attempted past the end of memory an OVF_ERR interrupt will be generated. If RDY/BSYB is violated, then data will ignored and an OVF_ERR interrupt will be generated. Once the SPI transaction has ended the ISD2100 will finish the flash write operation. When this operation is complete the ISD2100 will generate a WR_FIN interrupt. While device is actively writing to flash memory the CMD_BSY bit will be active.

Digital and Audio commands cannot occur concurrently. To prevent possibility of data corruption, user should disable any Voice Macro triggers and check status to ensure device is idle (CMD_BSY=0) before sending a digital write command.

### 13.3.3 ERASE_MEM – Sector Erase Memory

| ERASE_MEM | | | | | |
|---|---|---|---|---|---|
| Byte Sequence: | Host controller | 0x24 | SA[23:16] | SA[15:8] | SA[7:0] |
| | ISD2100 | Status | Status | Status | Status |
| Description: | Erases one sector of memory starting from SA | | | | |
| Interrupt Generation: | ADDR_ERR if memory protected. CMD_ERR if device is busy. CMD_FIN when erase operation complete. | | | | |

This erases memory from the sector containing start address SA. The minimum erase block of internal memory is a 1KByte sector.

The command will be accepted if status bits PD=0, DBUF_RDY=1, VM_BSY=0, CBUF_FUL=0 and CMD_BSY=0. If any of these conditions are not met then a CMD_ERR interrupt will be generated and the command ignored. If memory is write-protected an ADDR_ERR interrupt is generated. Upon completion of erase a CMD_FIN interrupt is generated.

While the device is erasing no other commands will execute. If a PLAY is sent it is queued in the command buffer and will not execute until the erase is finished. If a DIG_RD or DIG_WR command is sent to the device, RDY/BSYB pin will hold off any data transfer until the ERASE_MEM has completed.

When ERASE_MEM is in progress, the Status bit 0 CMD_BSY goes high. Users could poll the status to see if the erasing is done.

Digital and Audio commands cannot occur concurrently. To prevent possibility of data corruption, user should disable any Voice Macro triggers and check status to ensure device is idle (CMD_BSY=0) before sending a digital erase command.

### 13.3.4 CHIP_ERASE

| CHIP_ERASE | | | |
|---|---|---|---|
| Byte Sequence: | Host controller | 0x26 | 0x01 |
| | ISD2100 | Status Byte | Status Byte |
| Description: | Initiate a mass erase of memory. | | |
| Interrupt Generation: | CMD_ERR if device is busy and cannot accept command. CMD_FIN when erase operation complete. | | |

This erases the entire contents of the flash memory.

The command will be accepted if status bits PD=0, DBUF_RDY=1, VM_BSY=0, CBUF_FUL=0 and CMD_BSY=0. If any of these conditions are not met, then a CMD_ERR interrupt will be generated and the command ignored. If memory is mass erase protected an ADDR_ERR interrupt is generated. Upon completion of erase a CMD_FIN interrupt is generated.

While the device is erasing no other commands will execute. If a PLAY is sent it is queued in the command buffer and will not execute until the erase is finished. If a DIG_RD or DIG_WR command is sent to the device, RDY/BSYB pin will hold off any data transfer until the CHIP_ERASE has completed.

When CHIP_ERASE is in progress, the Status bit 0 CMD_BSY goes high. Users could poll the status to see if the erasing is done.

### 13.3.5 CHECKSUM

| CHECKSUM | | | | | |
|---|---|---|---|---|---|
| Byte Sequence: | Host controller | 0xF2 | EA[23:16] | EA[15:8] | EA[7:0] |
| | ISD2100 | Status Byte | Status | Status | Status |
| Description: | Initiate a checksum of memory. | | | | |
| Interrupt Generation: | CMD_ERR if device is busy and cannot accept command. CMD_FIN when erase operation complete. | | | | |

This initiates a 4-byte checksum calculation from the very beginning to the specified end address. The calculated checksum is stored in configuration registers 0x10 – 0x013. To re-calculate the checksum with a different end address, users have to write register 1 followed by 0 to CFG_REG4 bit-4 to clear the registers 0x10 – 0x13.

The command will be accepted if status bits PD=0, DBUF_RDY=1, VM_BSY=0, CBUF_FUL=0 and CMD_BSY=0. If any of these conditions are not met then a CMD_ERR interrupt will be generated and the command ignored. If memory is mass erase protected an ADDR_ERR interrupt is generated. Upon completion of erase a CMD_FIN interrupt is generated.

When CHECKSUM is in progress, the Status bit 0 CMD_BSY goes high. Users may poll CMD_BSY to determine if the checksum calculation is complete.

## 13.4 DEVICE CONFIGURATION COMMANDS.

This section describes 6 commands used to configure the ISD2100. These commands are used to:

- Set up the clocking regime of the device including clock source and setting the master sample rate.
- Configure the audio signal path.

The signal path, compression and sample rate configuration are controlled by forty-eight bytes of configuration register. These forty-eight bytes can be written individually or in a continuous sequential manner.

### 13.4.1 PWR_UP – Power up

| PWR_UP | | | | |
|---|---|---|---|---|
| Byte Sequence: | Host controller | 0x10 | | |
| | ISD2100 | Status | | … |
| Description: | Powers up device and initiates the power up sequence. | | | |

This command powers up the device. If device already powered up this command has no effect. If powered down, then the internal power up sequence is initiated. If the PU voice macro is present this is executed, otherwise the device defaults to power up the internal oscillator. When power up is complete the PD bit of the status register will go low and the RDY bit high. Until this event no other commands will be accepted by the ISD2100.

A formal power-up procedure is as follows:

- Send PWR_UP command.
- Poll Status until bit-6 DBUF_RDY goes high, which means ready.
- Poll Status until bit-2 VM_BSY goes low, which means voice macro 1 finishes.

### 13.4.2 PWR_DN – Power Down

| PWR_DN | | | | |
|---|---|---|---|---|
| Byte Sequence: | Host controller | 0x12 | | |
| | ISD2100 | Status | | … |
| Description: | Powers down the device after any active commands finish | | | |

This command powers down the device. If the device is currently executing a command the device will powers down when the command finishes. If playing or executing a voice macro, device will power down after playback is finished. The PWR_DN command will not generate an interrupt. PWR_DN has executed when PD bit of status goes high.

### 13.4.3 Reset

| RESET | | | | |
|---|---|---|---|---|
| Byte Sequence: | Host controller | 0x14 | | |
| | ISD2100 | Status | | … |
| Description: | Reset command resets all the registers and initiates POI procedure | | | |

The Reset command is one byte SPI command which resets all the registers and initiates a POI execution, if there is POI VM0 available, if the POI Voice Macro is empty the device will go to power down.

### 13.4.4 SET_CLK_CFG – Set Clock Configuration Register

| SET_CLK_CFG | | | | |
|---|---|---|---|---|
| Byte Sequence: | Host controller | 0xB4 | CFG_CLK[7:0] | |
| | ISD2100 | Status Byte | Status Byte | |
| Description: | Loads clock configuration register. | | | |

This sets the clock configuration register. The part reconfigures the clock and PLL configuration and waits for stable clock conditions before accepting new commands. When the configuration is changed, CMD_BSY will go high until clock configuration is complete. No new commands should be sent when the device status shows that device is busy. This command does not generate an interrupt.

### 13.4.5 RD_CLK_CFG – Read Clock Configuration Register

| | |
|---|---|
| Description: | Reads clock configuration register. |

This reads the clock configuration register.

| **RD_CLK_CFG** | | | | |
|---|---|---|---|---|
| Byte Sequence: | Host controller | 0xB6 | 0xXX | |
| | ISD2100 | Status Byte | CFG_CLK[7:0] | |
| Description: | Reads clock configuration register. | | | |

### 13.4.6 WR_CFG_REG – Write Configuration Register

| **WR_CFG_REG** | | | | | | |
|---|---|---|---|---|---|---|
| Byte Sequence: | Host controller | 0xB8 | REG[7:0] | D0 | … | Dn |
| | ISD2100 | STATUS0 | | | … | |
| Description: | Loads configuration register CFG[REG] with D0. Data bytes 1..n can be sent to load CFG[REG+1] with D1 to CFG[REG+n] with Dn. | | | | | |

This command loads configuration registers starting at the address specified. If multiple data bytes are sent, additional configuration registers are loaded. See Section 12.3 for details on configuration registers. There are forty-eight configuration registers in the ISD2100, REG0 – REG2F.

### 13.4.7 RD_CFG_REG – Read Configuration Register

| **RD_CFG_REG** | | | | | | |
|---|---|---|---|---|---|---|
| Byte Sequence: | Host controller | 0xBA | REG[7:0] | X | … | X |
| | ISD2100 | STATUS0 | | D0 | … | Dn |
| Description: | Reads configuration register CFG[REG] and outputs to SPI as D0. Data bytes 1..n can be read sequentially from CFG[REG+1] to CFG[REG+n]. | | | | | |

This command reads the configuration register starting at the address specified. If multiple data bytes are sent, additional configuration registers are read.

See Section 12.3 for details on configuration registers.

# 14 ELECTRICAL CHARACTERISTICS

## 14.1 OPERATING CONDITIONS

| DESCRIPTION | SYMBOL | CONDITION | MIN | MAX | UNITS |
|---|---|---|---|---|---|
| DC Power Supply | $V_{CCD}$ | $V_{CCD}$ - $V_{SSD}$ | -0.3 | +4.0 | V |
| | $V_{CCPCM}$ | $V_{CCPCM}$ – $V_{SSPCM}$ | -0.3 | +4.0 | V |
| Digital Input Voltage | $DV_{IN}$ | $DV_{IN}$ - $V_{SSD}$ | $V_{SSD}$ – 0.3 | $V_{CCD}$ + 0.3 | V |
| Junction Temperature | $T_J$ | - | -40 | +125 | °C |
| Storage Temperature | $T_{st}$ | - | -65 | +150 | °C |

CAUTION: Do not operate at or near the maximum ratings listed for extended period of time. Exposure to such conditions may adversely influence product reliability and result in failures not covered by warranty. These devices are sensitive to electrostatic discharge; follow proper IC Handling Procedures.

## 14.2 OPERATING CONDITIONS

### OPERATING CONDITIONS (INDUSTRIAL PACKAGED PARTS)

| CONDITIONS | VALUES |
|---|---|
| Operating temperature range (Case temperature) | -40°C to +85°C |
| Supply voltage ($V_{DD}$) [1] | +2.7V to +3.6V |
| Ground voltage ($V_{SS}$) [2] | 0V |
| Digital Input voltage ($DV_{IN}$) | 0V to 3.6V |
| Voltage applied to any pins | ($V_{SS}$ –0.3V) to ($V_{DD}$ +0.3V) |

NOTES: [1] $V_{DD}$ = $V_{CCD}$ = $V_{CCPWM}$

[2] $V_{SS}$ = $V_{SSD}$ = $V_{SSPWM}$

## 14.3 AC PARAMETERS

### 14.3.1 Internal Oscillator

| PARAMETER | SYMBOL | MIN | TYP | MAX | UNITS | CONDITIONS |
|---|---|---|---|---|---|---|
| Sample rate with Internal Oscillator | $F_{Smax}$ | -1% | 32kHz | +1% | kHz | Vdd = 3V.<br>At room temperature |

### 14.3.2 Speaker Outputs

| PARAMETER | SYMBOL | MIN | TYP[1] | MAX | UNITS | CONDITIONS |
|---|---|---|---|---|---|---|
| SNR, Memory to SPK+/SPK- | $SNR_{MEM\_SPK}$ | | 60 | | dB | Load 150Ω [2][3] |
| Output Power | $P_{OUT\_SPK}$ VCC=3.0 | | | 500 | mW | @3.0V, Load 4Ω [2] |
| | | | | 630 | mW | @3.3V, Load 4Ω [2] |
| | | | | 330 | mW | @3.0V, Load 8Ω [2] |
| | | | | 420 | mW | @3.3V, Load 8Ω [2] |
| THD, Memory to SPK+/SPK- | THD % | | <1% | | | Load 8Ω [2] |
| Minimum Load Impedance | $R_{L(SPK)}$ | 4 | 8 | | Ω | |

Notes: [1] Conditions $V_{cc}$=3V, $T_A$=25°C unless otherwise stated.
[2] Based on 12-bit PCM.
[3] All measurements are C-message weighted.

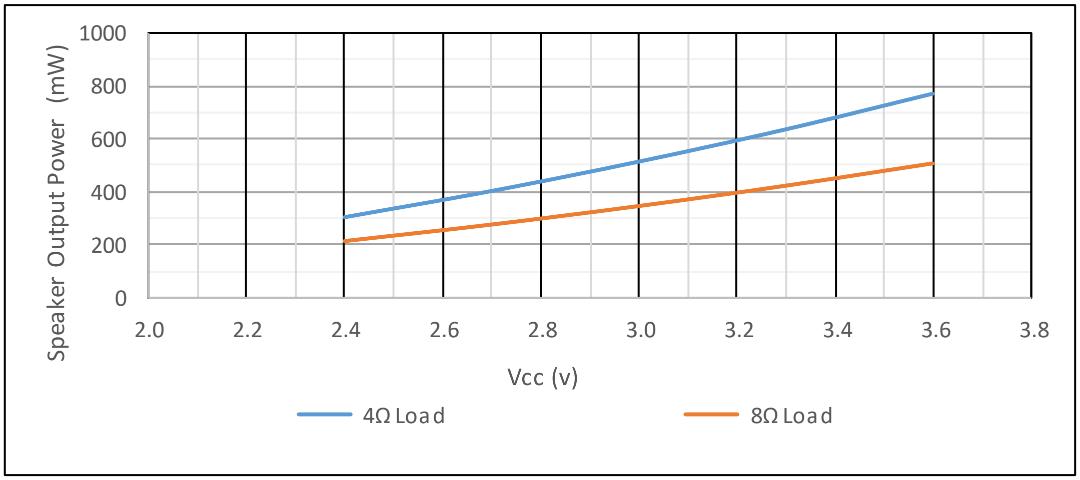



Figure 14-1 Speaker output power

## 14.4 DC Parameters

| PARAMETER | SYMBOL | MIN | TYP [1] | MAX | UNITS | CONDITIONS |
|---|---|---|---|---|---|---|
| Supply Voltage | $V_{DD}$ | 2.7 | | 3.6 | V | |
| Input Low Voltage | $V_{IL}$ | $V_{SS}$-0.3 | | 0.3x$V_{DD}$ | V | |
| Input High Voltage | $V_{IH}$ | 0.7x$V_{DD}$ | | $V_{DD}$ | V | |
| Output Low Voltage | $V_{OL}$ | $V_{SS}$-0.3 | | 0.3x$V_{DD}$ | V | $I_{OL}$ = 1mA |
| Output High Voltage | $V_{OH}$ | 0.7x$V_{DD}$ | | $V_{DD}$ | V | $I_{OH}$ = -1mA |
| Pull-up Resistance | $R_{PU}$ | | 50 | | kΩ | |
| Pull-down Resistance | $R_{PD}$ | | 10 | | kΩ | |
| INTB Output Low Voltage | $V_{OH1}$ | | | 0.4 | V | |
| Playback Current | $I_{DD\_Playback}$ | | 3 | | mA | No Load [2] |
| Standby Current | $I_{SB}$ | | <1 | 10 | µA | $V_{DD}$= 3.6V |
| Input Leakage Current | $I_{IL}$ | | | ±1 | µA | Force $V_{DD}$ |

Notes: [1] Conditions $V_{DD}$=3V, $T_A$=25°C unless otherwise stated

[2] To calculate total current, add load dissipation into application specific load.

## 14.5 SPI Timing

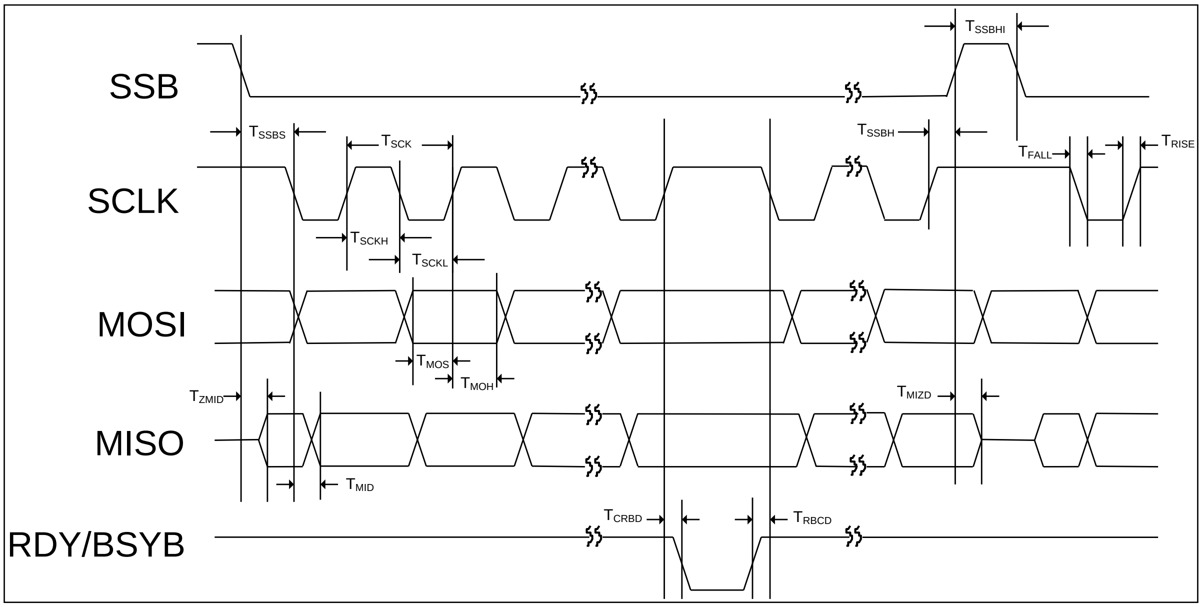



Figure 14-2 SPI Timing

| SYMBOL | DESCRIPTION | MIN | TYP | MAX | UNIT |
|---|---|---|---|---|---|
| $T_{SCK}$ | SCLK Cycle Time | 60 | --- | --- | ns |
| $T_{SCKH}$ | SCLK High Pulse Width | 25 | --- | --- | ns |
| $T_{SCKL}$ | SCLK Low Pulse Width | 25 | --- | --- | ns |
| $T_{RISE}$ | Rise Time | --- | --- | 10 | ns |
| $T_{FALL}$ | Fall Time | --- | --- | 10 | ns |
| $T_{SSBS}$ | SSB Falling Edge to 1st SCLK Falling Edge Setup Time | 30 | --- | --- | ns |
| $T_{SSBH}$ (Note 1) | Last SCLK Rising Edge to SSB Rising Edge Hold Time (ISD2130 Only) | 30ns | --- | 50us | --- |
| $T_{SSBH}$ (Note 1) | Last SCLK Rising Edge to SSB Rising Edge Hold Time (ISD2115A only) | 30ns | --- | 10us | --- |
| $T_{SSBHI}$ | SSB High Time between SSB Lows | 20 | --- | --- | ns |
| $T_{MOS}$ | MOSI to SCLK Rising Edge Setup Time | 15 | --- | --- | ns |
| $T_{MOH}$ | SCLK Rising Edge to MOSI Hold Time | 15 | --- | --- | ns |
| $T_{ZMID}$ | Delay Time from SSB Falling Edge to MISO Active | -- | -- | 12 | ns |
| $T_{MIZD}$ | Delay Time from SSB Rising Edge to MISO Tri-state | -- | -- | 12 | ns |
| $T_{MID}$ | Delay Time from SCLK Falling Edge to MISO | --- | --- | 12 | ns |
| $T_{CRBD}$ | Delay Time: SCLK Rising Edge to RDY/BSYB Falling Edge | -- | -- | 12 | ns |
| $T_{RBCD}$ | Delay Time: RDY/BSYB Rising Edge to SCLK Falling Edge | 0 | -- | -- | ns |

Note 1: It is suggested to use 10µs as max for $T_{SSBH}$ to ensure software compatibility with ISD2115A and ISD2130.

## 15 APPLICATION DIAGRAM

The following applications example is for reference only. It makes no representation or warranty that such applications shall be suitable for the use specified. Each design has to be optimized in its own system for the best performance on voice quality, current consumption, functionality etc.

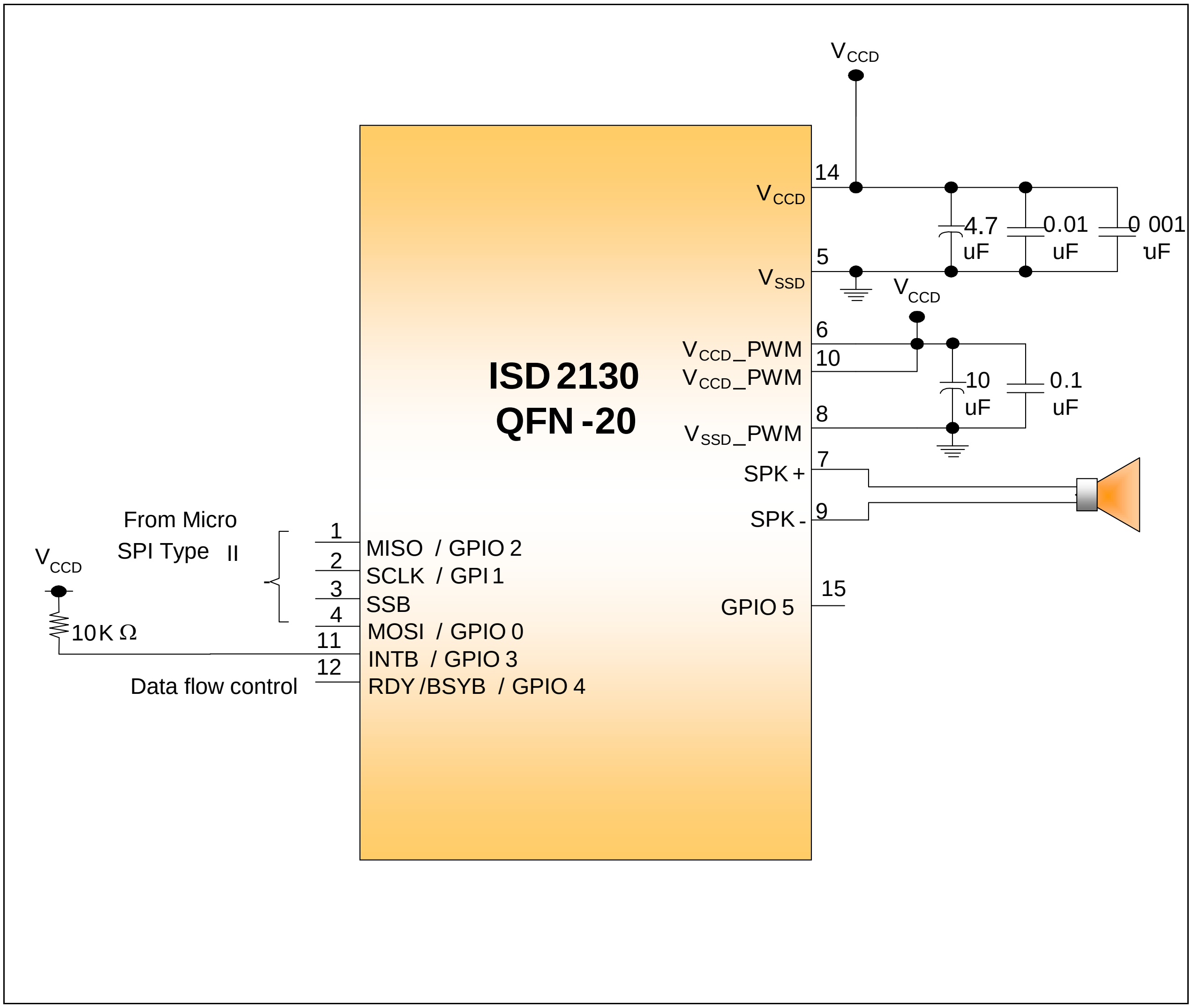



Figure 15-1 ISD2100 Application Diagram Example for programming with a Microcontroller SPI Mode

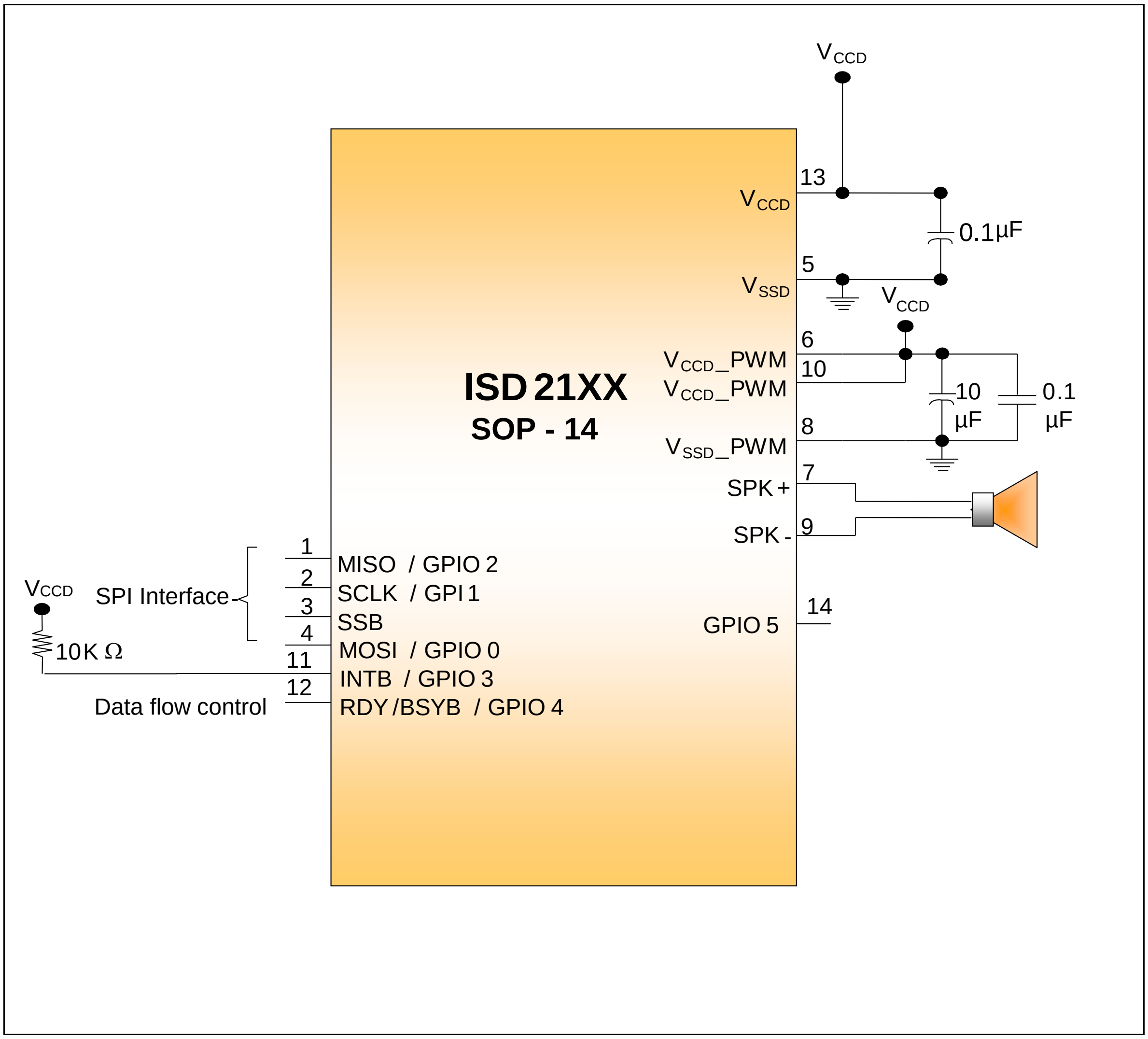



Figure 15-2 ISD2100 Application Diagram Example for programming with a Microcontroller SPI Mode

The following applications example is for reference only. It makes no representation or warranty that such applications shall be suitable for the use specified. Each design has to be optimized in its own system for the best performance on voice quality, current consumption, functionality etc.

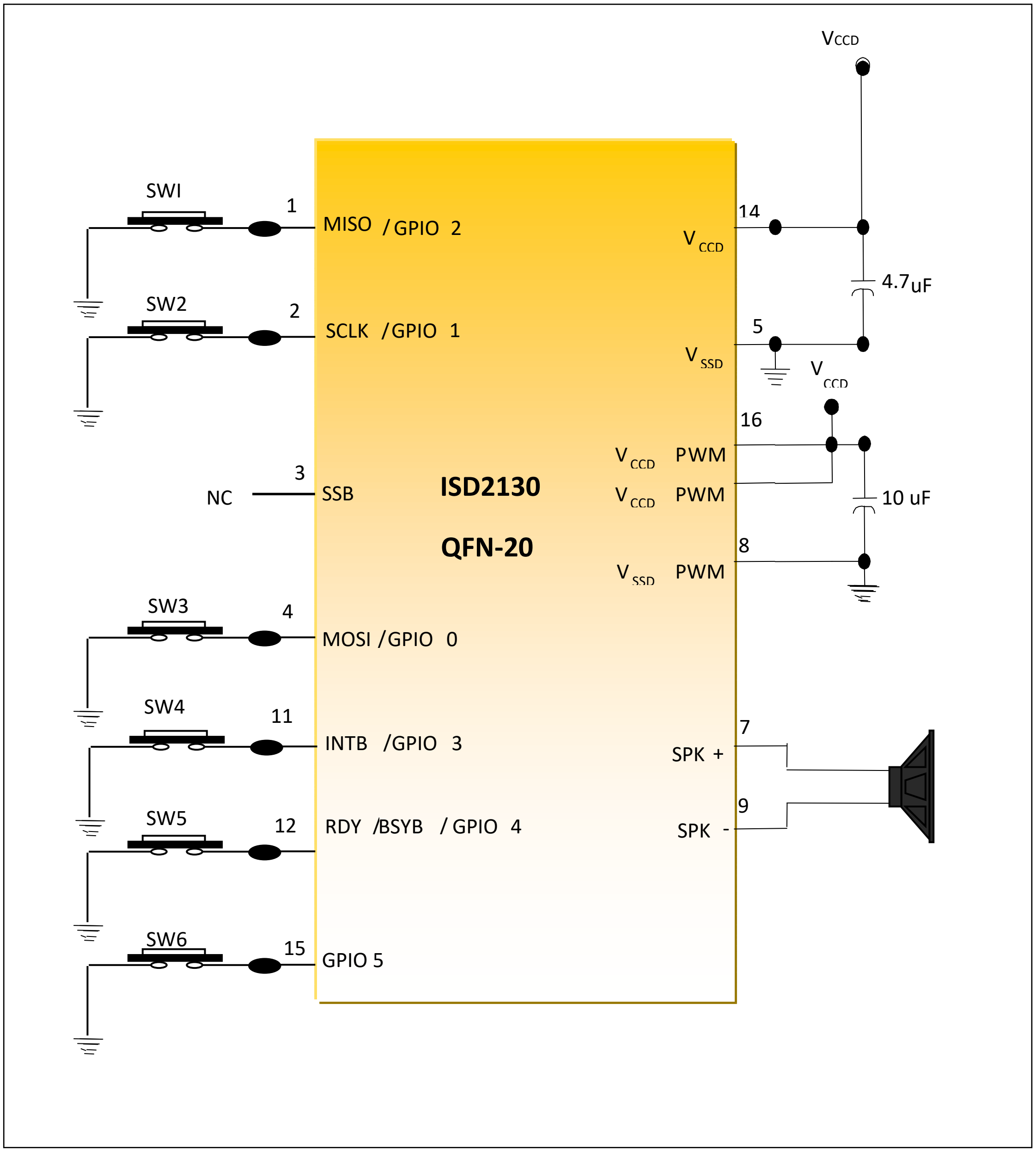



Figure 15-3 ISD2100 Application Diagram Stand-Alone Mode

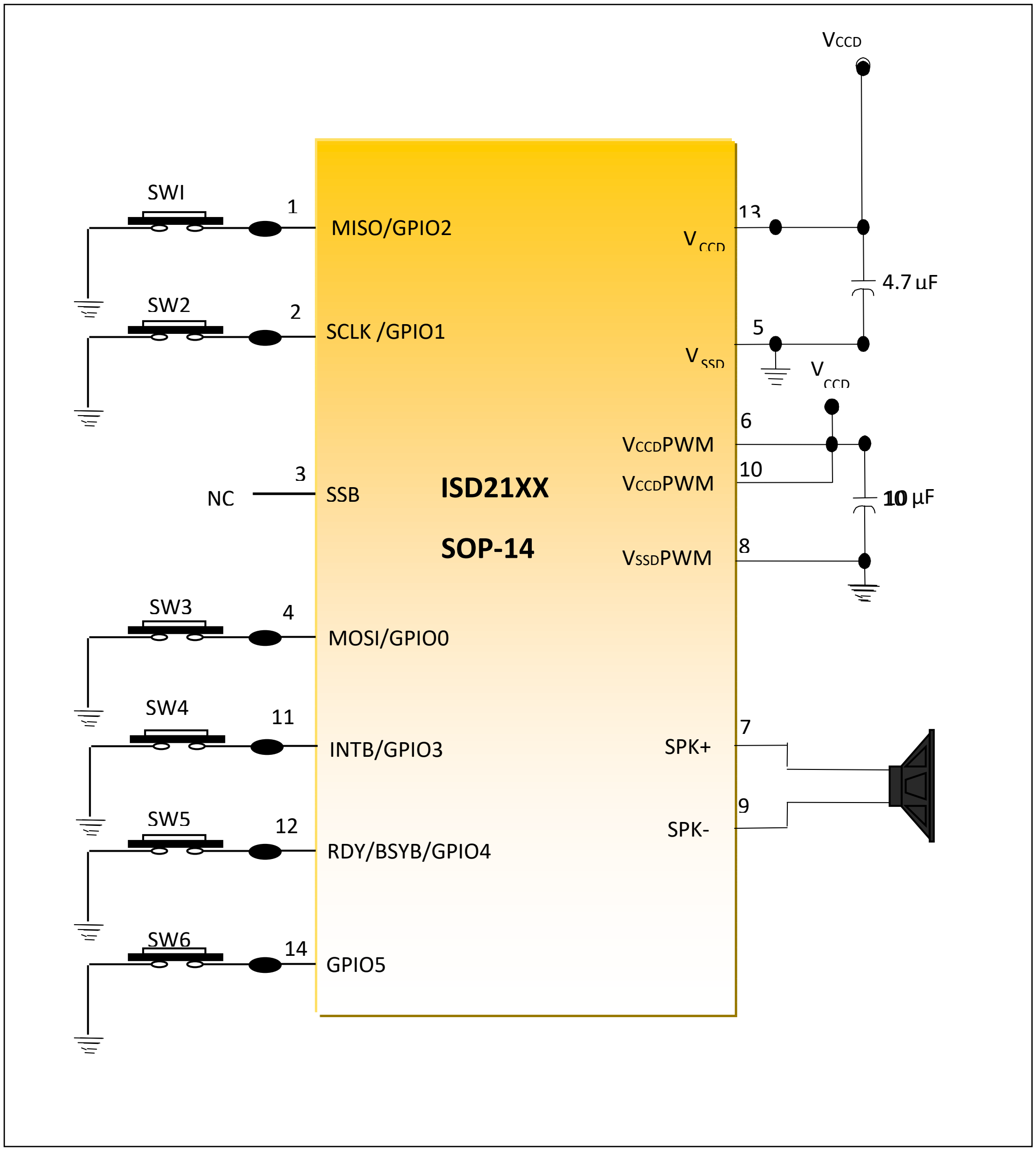
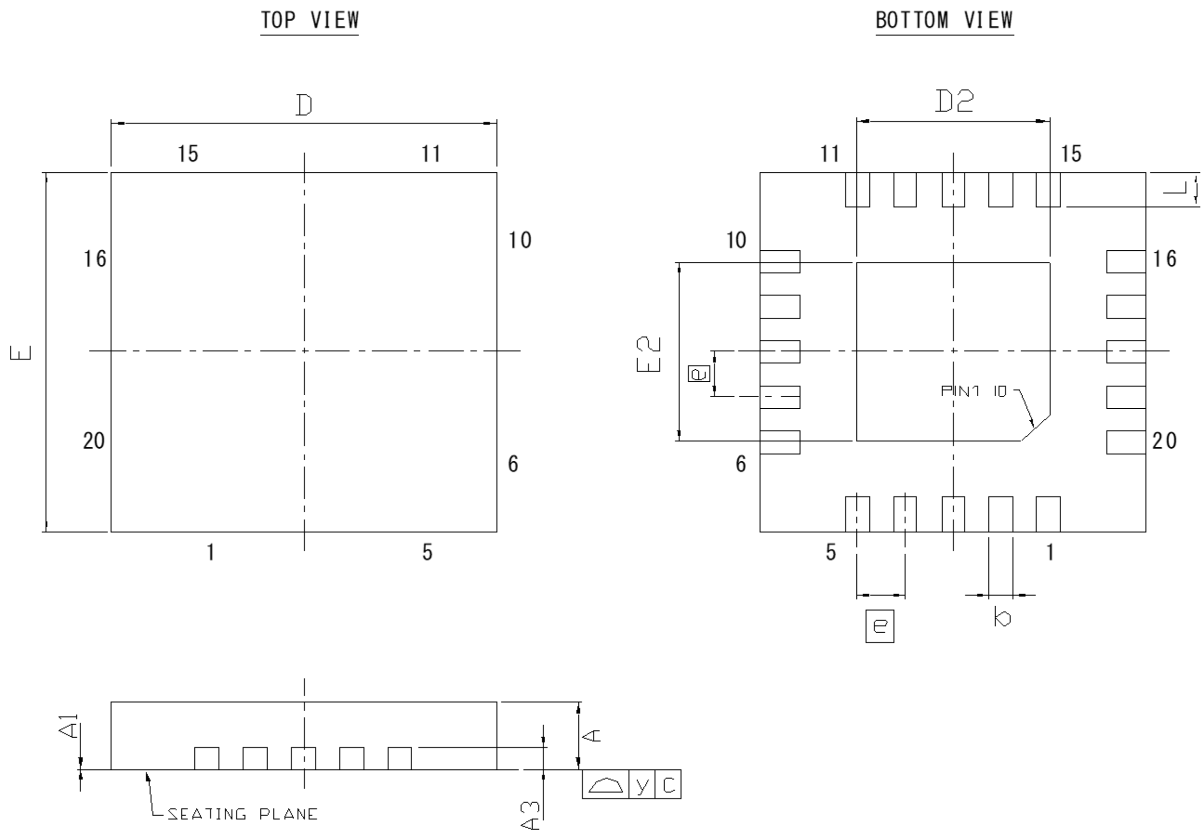



Figure 15-4 ISD2100 Application Diagram Stand-Alone Mode

# 16 PACKAGE SPECIFICATION

## 16.1 20 Lead QFN

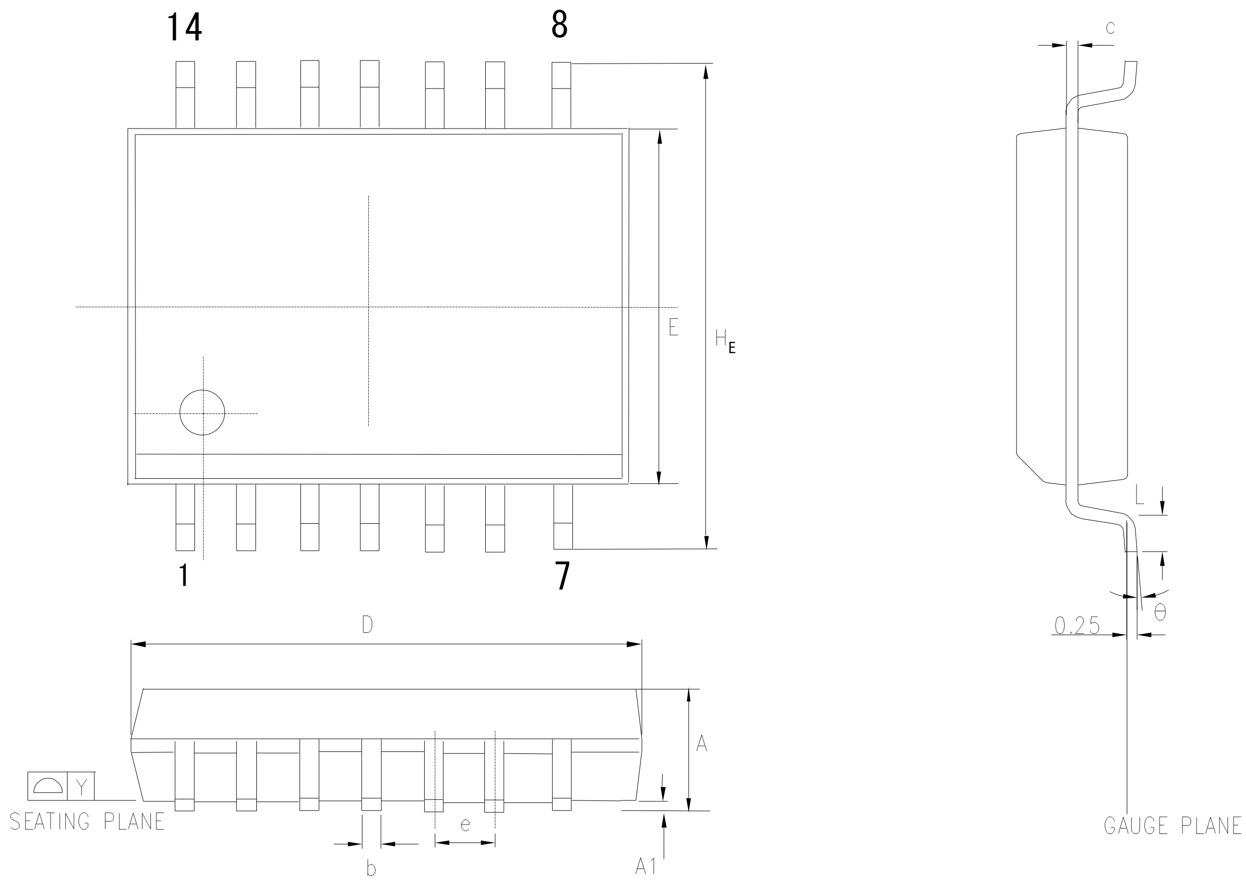



Controlling Dimension :Millimeters

| SYMBOL | DIMENSION (MM) | | | DIMENSION (Inch) | | |
|---|---|---|---|---|---|---|
| | MIN. | NOM. | MAX. | MIN. | NOM. | MAX. |
| A | 0.70 | 0.75 | 0.80 | 0.02756 | 0.02953 | 0.03150 |
| A1 | 0 | 0.02 | 0.05 | 0 | 0.0079 | 0.00197 |
| A3 | 0.203 REF | | | 0.0079 REF | | |
| b | 0.18 | 0.25 | 0.30 | 0.00709 | 0.00984 | 0.01181 |
| D | 3.90 | 4.00 | 4.10 | 0.1535 | 0.1575 | 0.1614 |
| D2 | 1.90 | 2.00 | 2.10 | 0.0748 | 0.0787 | 0.0827 |
| E | 3.90 | 4.00 | 4.10 | 0.1535 | 0.1575 | 0.1614 |
| E2 | 1.90 | 2.00 | 2.10 | 0.0748 | 0.0787 | 0.0827 |
| e | 0.50 BSC | | | 0.01969 BSC | | |
| L | 0.30 | 0.40 | 0.50 | 0.01181 | 0.01574 | 0.01969 |
| y | 0.08 | | | 0.00315 | | |

Note.D2,E2 by die size difference .

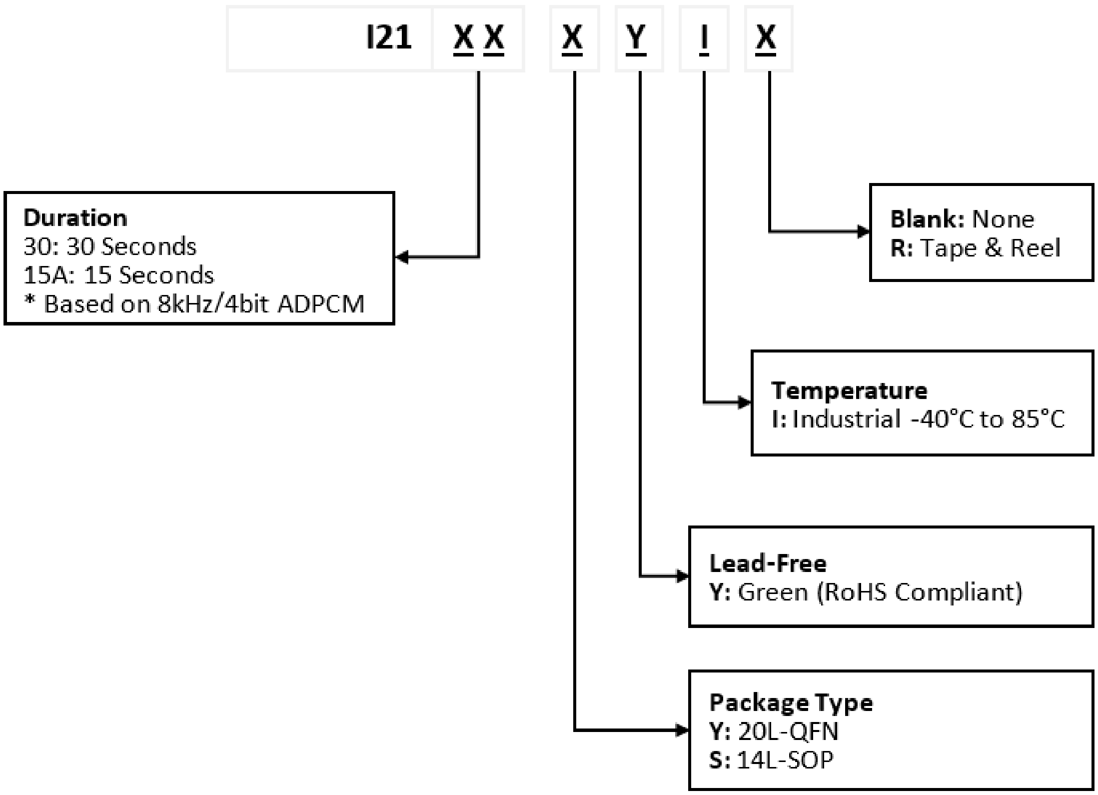
### 16.2 14 LEAD 150-MIL SMALL OUTLINE PACKAGE




Control demensions are in milmeters .

| SYMBOL | DIMENSION IN MM | | DIMENSION IN INCH | |
|---|---|---|---|---|
| | MIN. | MAX. | MIN. | MAX. |
| A | 1.35 | 1.75 | 0.053 | 0.069 |
| A1 | 0.10 | 0.25 | 0.004 | 0.010 |
| b | 0.33 | 0.51 | 0.013 | 0.020 |
| c | 0.19 | 0.25 | 0.008 | 0.010 |
| E | 3.80 | 4.00 | 0.150 | 0.157 |
| D | 8.55 | 8.75 | 0.337 | 0.344 |
| e | 1.27 BSC | | 0.050 BSC | |
| $H_E$ | 5.80 | 6.20 | 0.228 | 0.244 |
| Y | — | 0.10 | — | 0.004 |
| L | 0.40 | 1.27 | 0.016 | 0.050 |
| θ | 0 | 8 | 0 | 8 |

# 17 ORDERING INFORMATION




Ordering Information

| Part Number | Duration | Package | Temperature | Notes |
|---|---|---|---|---|
| **I2115AYYI** | 15s | 20L-QFN | -40°C to 85°C | |
| **I2115AYYIR** | 15s | 20L-QFN, Tape & Reel | -40°C to 85°C | |
| **I2115ASYI** | 15s | 14L-SOP | -40°C to 85°C | |
| **I2130YYI** | 30s | 20L-QFN | -40°C to 85°C | |
| **I2130YYIR** | 30s | 20L-QFN, Tape & Reel | -40°C to 85°C | |
| **I2130SYI** | 30s | 14L-SOP | -40°C to 85°C | |
| **I2130SYIR** | 30s | 14L-SOP, Tape & Reel | -40°C to 85°C | |

## 18 REVISION HISTORY

| Version | Date | Description |
|---|---|---|
| 1.0 | March 4, 2010 | Update description |
| 1.1 | April 01, 2010 | Update description |
| 1.2 | July 27, 2010 | Add 2110 duration |
| 1.3 | Oct 26, 2010 | Update description |
| 1.4 | Dec 03, 2010 | $I_{DD\_Playback}$ Update |
| 1.5 | June 16, 2011 | Add I2110A and I2115A |
| 1.6 | June 16, 2011 | Update SPI timing: TSSBH Maximum 10us for ISD2110A and ISD2115A |
| 1.7 | July 13, 2011 | Add 2115 duration |
| 1.8 | Aug 21, 2013 | Remove 2110A duration |
| 1.9 | July 13, 2016 | Add storage temperature |
| 2.0 | Aug 1, 2016 | Update Absolute Maximum Rating. Add $V_{CCD}$_PWM description. |
| 2.1 | Mar 30, 2020 | Update document format |
| 2.2 | May 20, 2020 | Add SOP-14 package and related description |
| 2.3 | July 24, 2020 | Fix Page 7, Package Drawing<br>Update Ordering information |
| 2.4 | May 6, 2021 | Update output power data |
| 2.5 | Jun 15, 2021 | Update Ordering Information |

## Important Notice

**Nuvoton Products are neither intended nor warranted for usage in systems or equipment, any malfunction or failure of which may cause loss of human life, bodily injury or severe property damage. Such applications are deemed, "Insecure Usage".**

**Insecure usage includes, but is not limited to: equipment for surgical implementation, atomic energy control instruments, airplane or spaceship instruments, the control or operation of dynamic, brake or safety systems designed for vehicular use, traffic signal instruments, all types of safety devices, and other applications intended to support or sustain life.**

**All Insecure Usage shall be made at customer's risk, and in the event that third parties lay claims to Nuvoton as a result of customer's Insecure Usage, customer shall indemnify the damages and liabilities thus incurred by Nuvoton.**