IA61x
SHP-AN04 Rev 1.0

IA61x

Initialization Application Note

Abstract

The IA61x is a combination of a PDM microphone and microprocessor. Thus, the power sequencing for the IA61x is different than that of a standard PDM microphone. The IA61x has a very low-power profile (more like a PDM microphone) and can take advantage of some of the power methods of a microphone that are not available to most microprocessors.

The IA61x is intended to be an always-on device. Once power has been applied, there is no need to remove it until the device powers off. The low-power states preserve battery power and have minimal impact on battery life.

The following general statement must be true: power (VDD_IO) must be applied before any of the six IO pins reach logic high. If an IO pin is driven high before power is applied, the IA61x can be powered from this IO pin and enter into an unstable state.

This document highlights the differences of the interfaces the IA61x supports and discusses the points to consider for each of these interfaces.

Purpose

This application note discusses the different configurations of the IA61x; it also provides helpful information for selecting the power source for the IA61x. It concludes with a discussion on error handling with the IA61x.

Target Audience

This document is for hardware, software, and system engineers interested in integrating the Knowles IA61x into their design.

Table of Contents

1. Pin Configuration

This section discusses each pin or bus to be considered during system design.

1.1 I²C

The I²C bus requires pull-up resistors. For the IA61x, these pull-up resistors must be connected to the VDD_IO rail that drives the IA61x (see Figure 1). This ensures that the IO rail is not driven before the IA61x is fully powered.

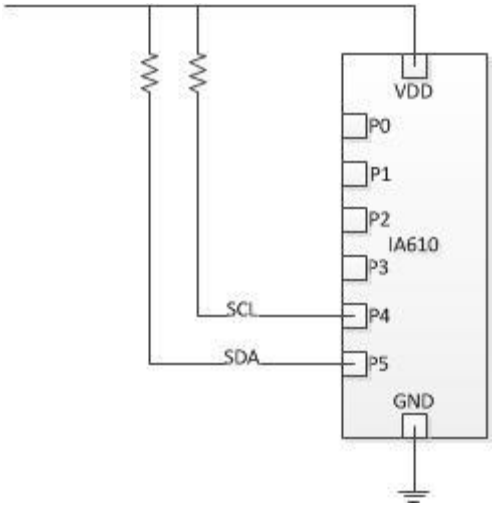

Figure 1 Pull-Up Resistors Configuration

For the system, this may require the IA61x either is on a dedicated I²C bus, or has a switch to disconnect these pins from the rest of the I²C bus during the power-up sequence.

A typical case is sharing I²C with the PMIC or Codec that is powering the IA61x. The I²C bus on the IA61x cannot be pulled high until VDD_IO is enabled, but communicating with the PMIC or Codec requires the I²C to be active. Knowles recommends either isolating the IA61x, sharing the I²C bus with some components, or using a switch to isolate the I²C bus.

1.1.1 IA61x I²C Isolation

Figure 2 shows the high-level connections when the I²C bus is required to enable the power rail to the IA61x. Usually, this is when the Codec is on I²C control; however, it also can be true for a PMIC or other power source.

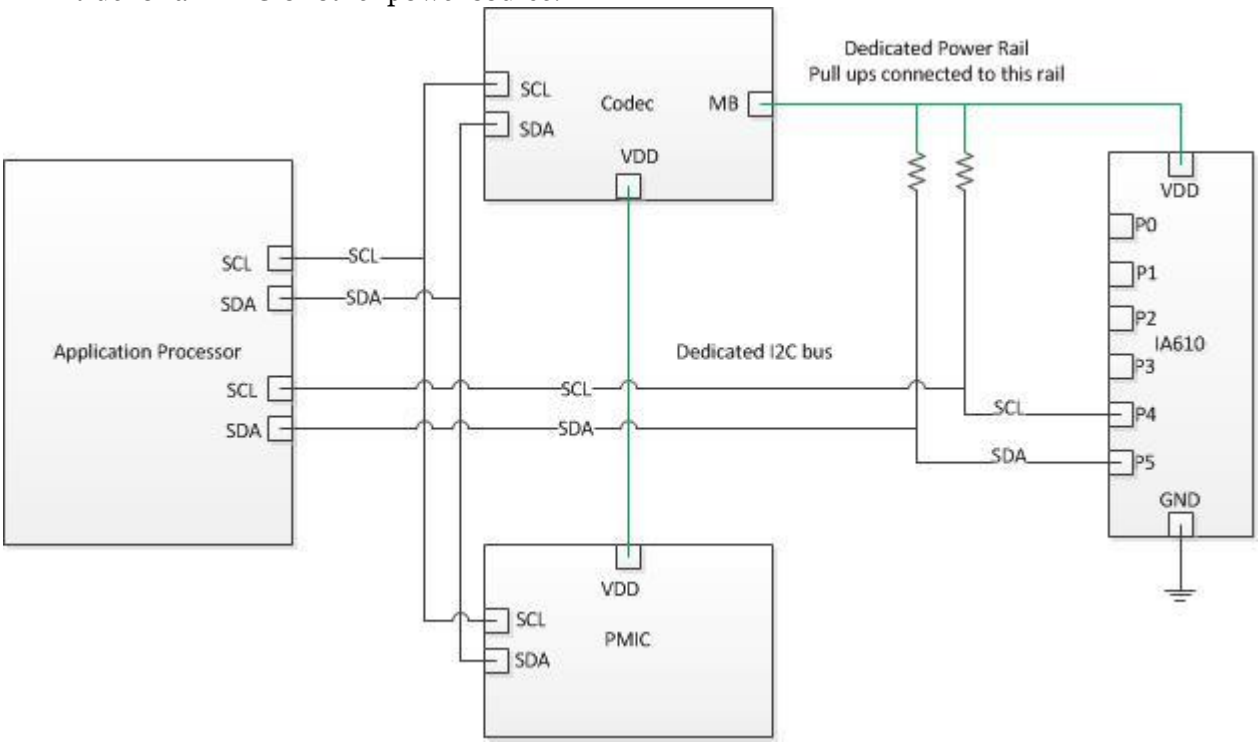

Figure 2 I²C Isolated Configuration

1.1.2 IA61x I²C Shared

Figure 3 shows an alternate configuration, where multiple components can share the I²C bus. In this mode, the power rail to the IA61x must be enabled before any traffic is running on the I²C bus.

The power rail can be shared across devices, as shown in Figure 3, but this is not required. It also is possible to have a dedicated power rail to the IA61x and only share the I²C bus with other devices.

While Figure 3 shows the I²C controlling the PMIC, other busses can be used to control it. For example, it is possible to use an SPI bus to configure the PMIC and the I²C to control the Codec and IA61x.

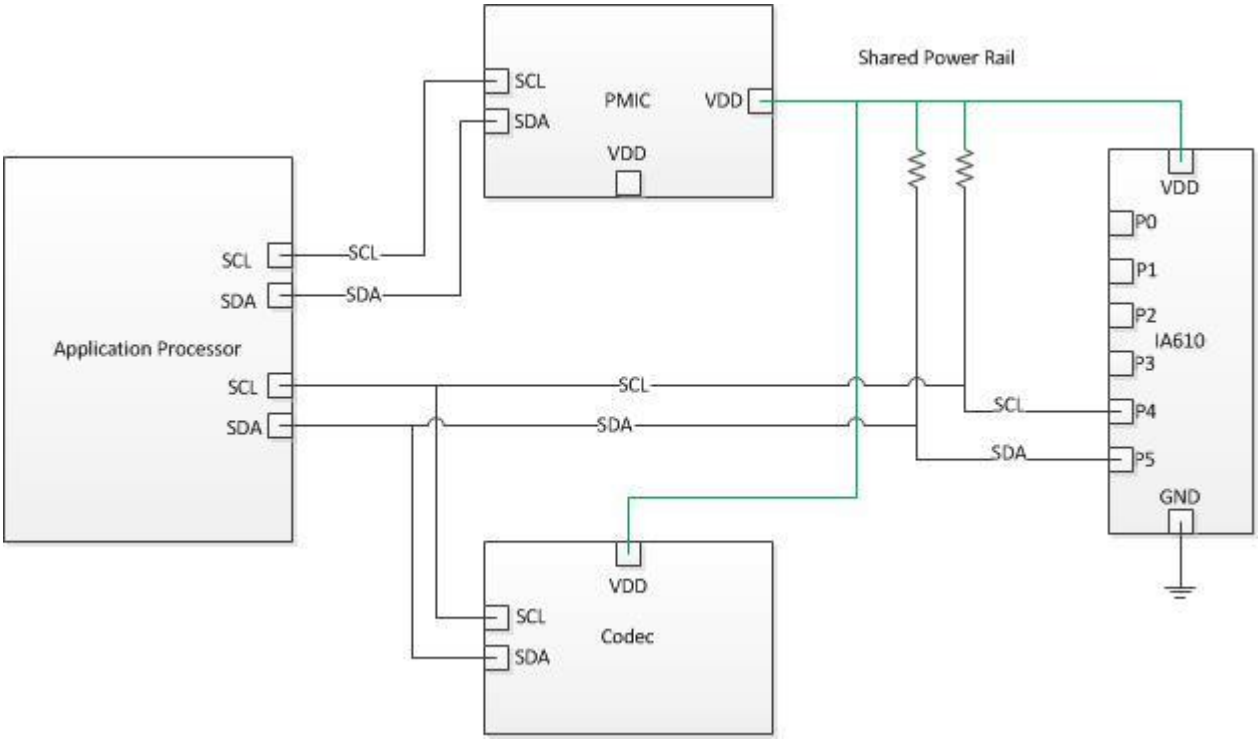

Figure 3 I²C Shared Configuration

1.1.3 IA61x Switch Configuration

Figure 4 shows a switch isolating the I²C bus on the IA61x during the power-up sequence. Once the sequence has been completed, a GPIO from the application processor connects the I²C bus.

In this configuration, the pull-up resistors on the I²C bus can be located anywhere on the bus. It is not required that pull-up resistors be connected to the VDD_IO rail powering the IA61x. Once the IA61x I²C bus is connected to the main I²C bus, the lines are pulled high.

If the designer places pull-up resistors as shown in Figure 4, it is important to keep in mind the total resistance for each line and not to exceed the I²C specification. Knowles recommends a very weak pull-up (100K) in this case.

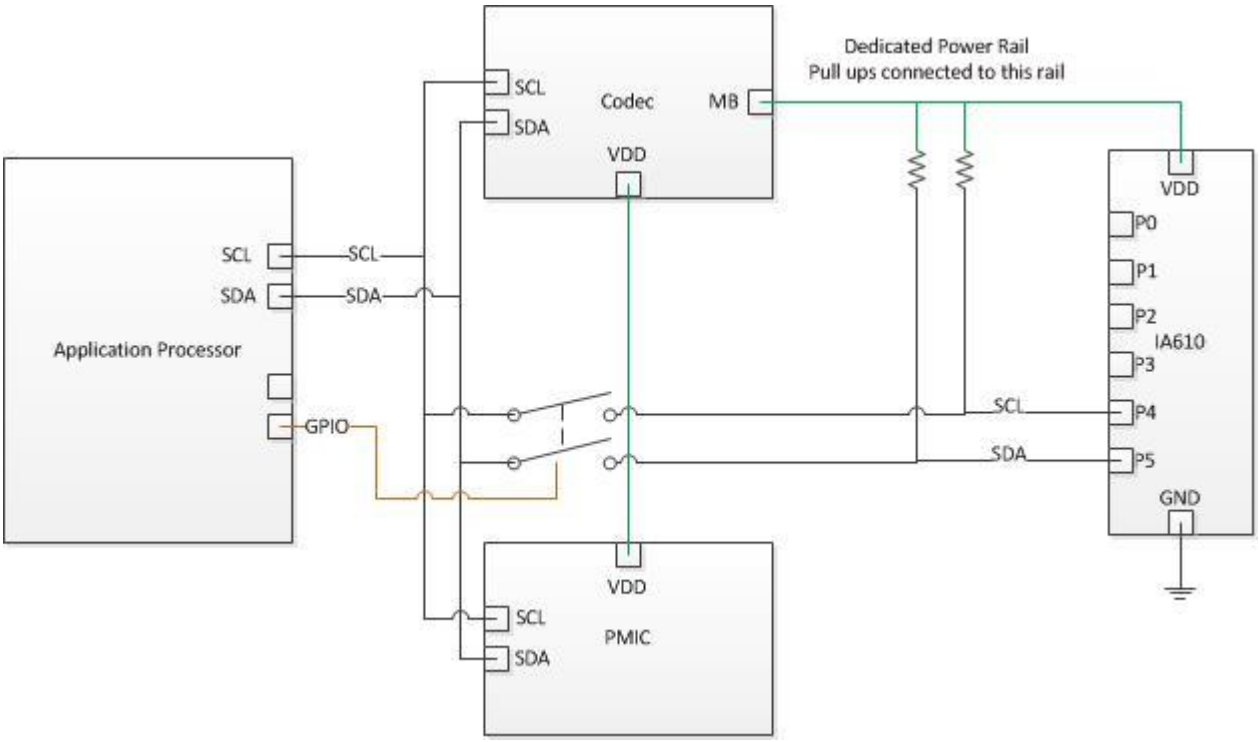

Figure 4 I²C Switch Configuration

1.2 UART

UART is a point-to-point bus with no impact on other components in the system (see Figure 5). When UART is enabled by the application processor, its normal state is to drive the pin high when there is no data. This must not happen until after the power (VDD_IO) is enabled to the IA61x.

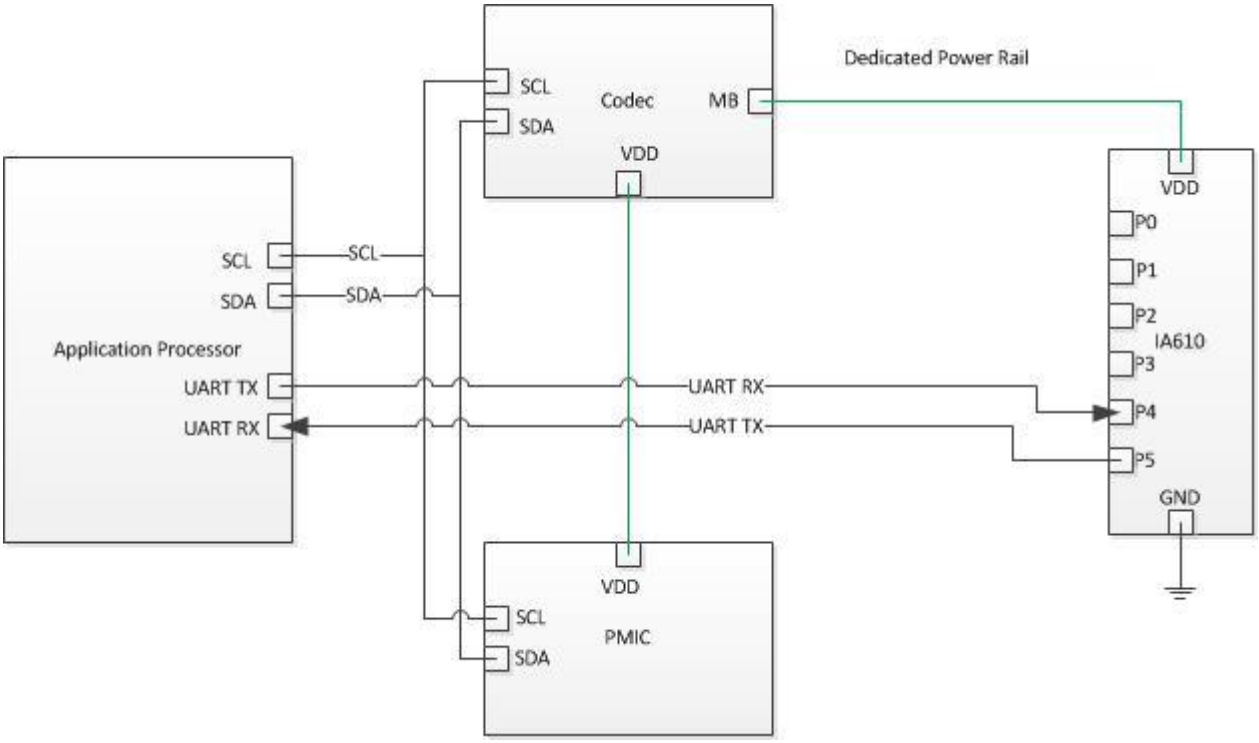

Figure 5 UART Configuration

If the UART TX line on the application processor cannot be held low during the power sequence, a switch is needed to isolate this signal during the power sequence.

1.3 SPI

The SPI bus can be shared between devices. It is highly recommended that no SPI traffic is generated until the power (VDD_IO) to the IA61x has been enabled.

SPI CS is an active-low configuration to boot the IA61x. This pin must be low until power is enabled.

In systems that cannot reconfigure the SPI CS, or if this is done before the power-up sequence is complete, an external switch is required to isolate the CS during the power-up sequence (see Figure 6). A dedicated GPIO is needed to control the switch on the SPI_CS line.

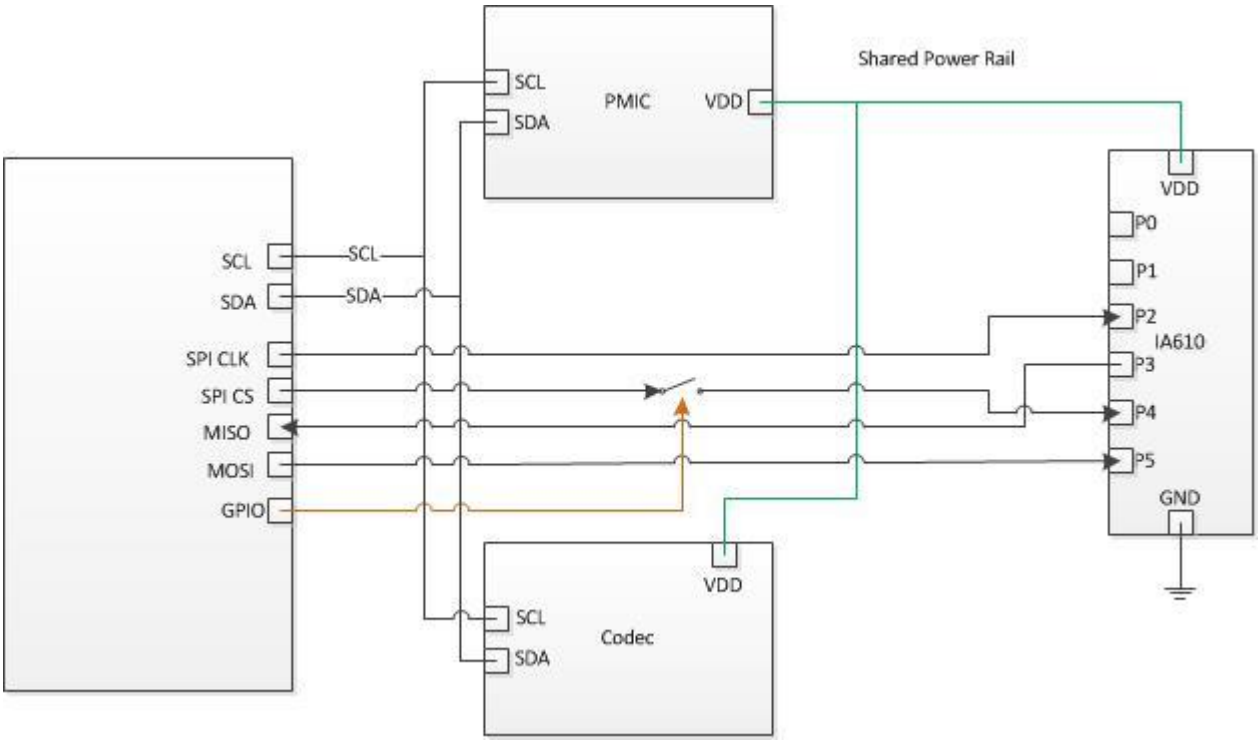

Figure 6 SPI CS Isolation Diagram

1.4 PDM

PDM can be in both a master and a slave configuration. In the master mode, the IA61x drives the PDM clock and receives PDM data. In the slave mode, the IA61x receives the PCM clock and drives PDM data on one edge per the datasheet.

1.4.1 PDM Master

This mode has no design concerns because the clock is driven from the IA61x. The assumption is that the PDM data source (the microphone) does not drive the line high until a

PDM clock is provided. If this cannot be done, an external switch is required to isolate the signal from the IA61x during the power-up sequence.

1.4.2 PDM Slave

The PDM clock must not be driven high until the IA61x power is enabled. The IA61x cannot drive the line until the power-up sequence is complete.

1.5 I²S

I²S traffic must be kept logic low until after the power-up sequence has completed.

1.6 GPIO

GPIO pins in I²C and UART modes must be kept logic low until after the power-up sequence has completed. If this cannot be done, an external switch is required to isolate the signal from the IA61x during the power-up sequence.

1.7 Latch-On Reset or Latch-On Power

Latch-on reset (LOR) pins have the same condition as the I²C: any pull up resistors must be connected to the rail powering the IA61x VDD_IO. In this configuration, as the power rail reaches the power good state, the pull-up resistor provides the positive signal to latch the state once the reset is de-asserted in the IA61x (see Figure 7).

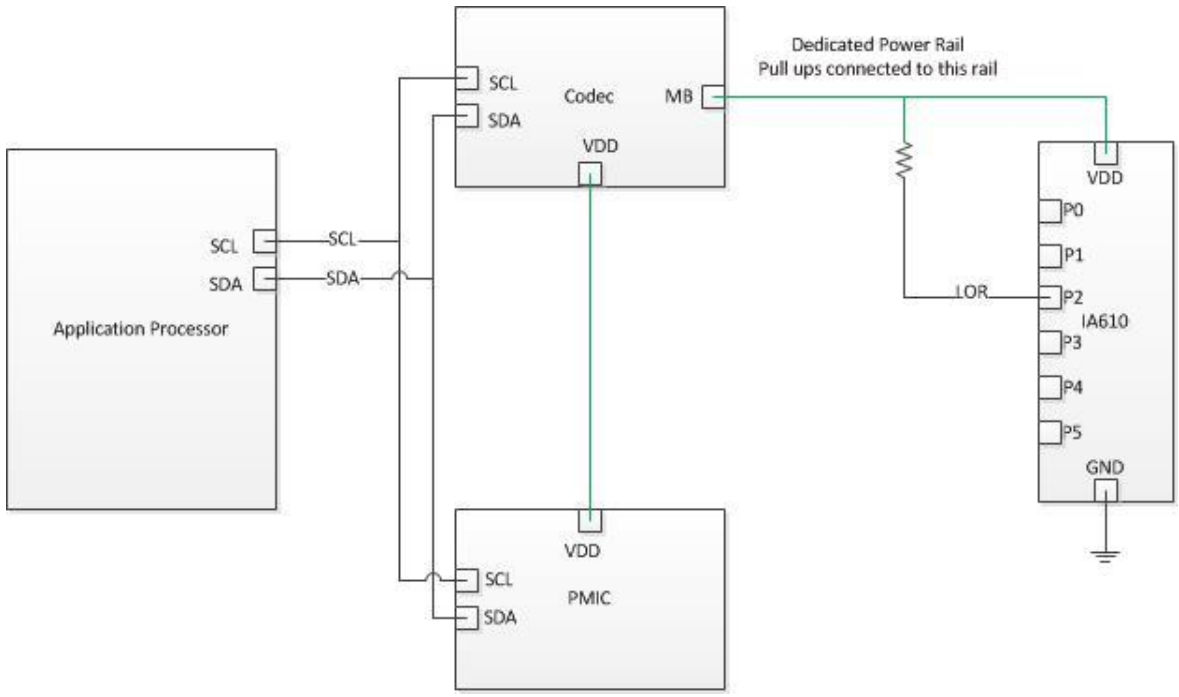

Figure 7 Latch-On Reset with Dedicated Power Supply

The LOR pins can be connected to a common power supply, as shown in Figure 8. This satisfies the condition that the pin must be kept low until the power is provided.

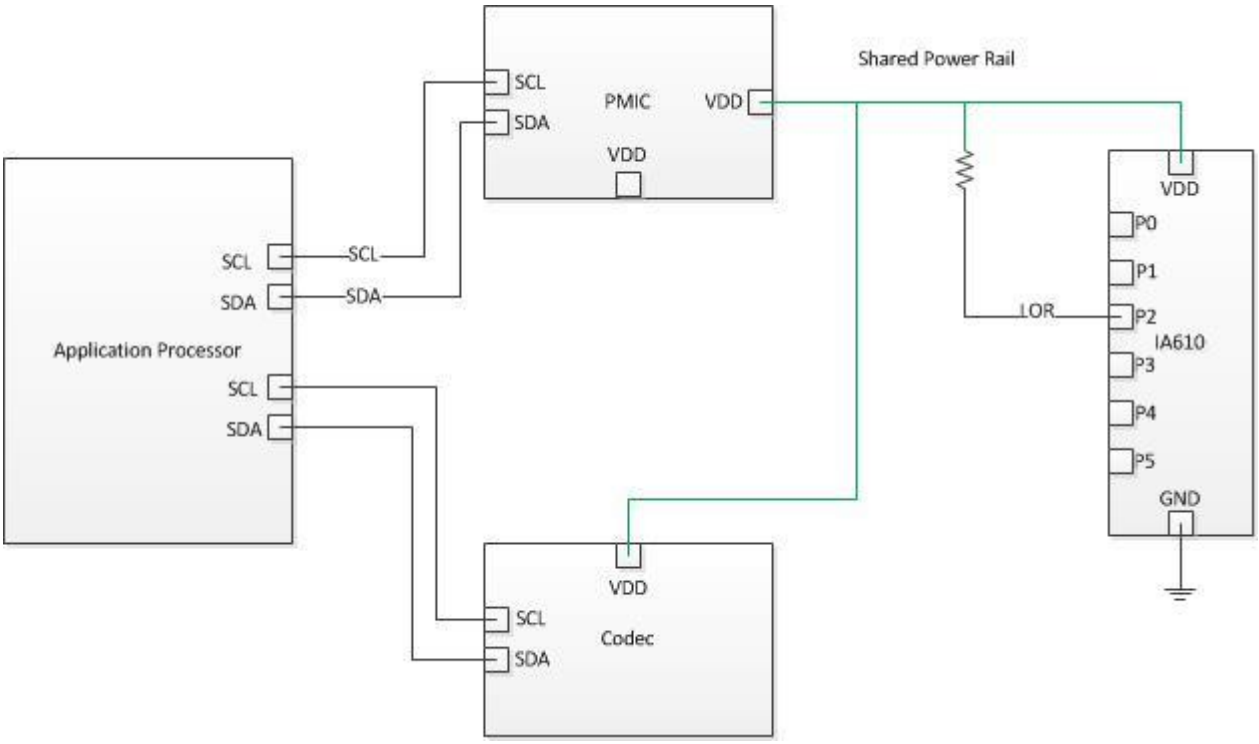

Figure 8 Latch On Reset with Shared Power Supply

Figure 9 shows the incorrect method for connecting a latch-on reset option: the LOR can be pulled high before the power is enabled on the IA61x, which is a faulty configuration. Also, this method does not work if the power rail on IA61x is enabled first, as the LOR is not in the right state to latch during the power-up sequence.

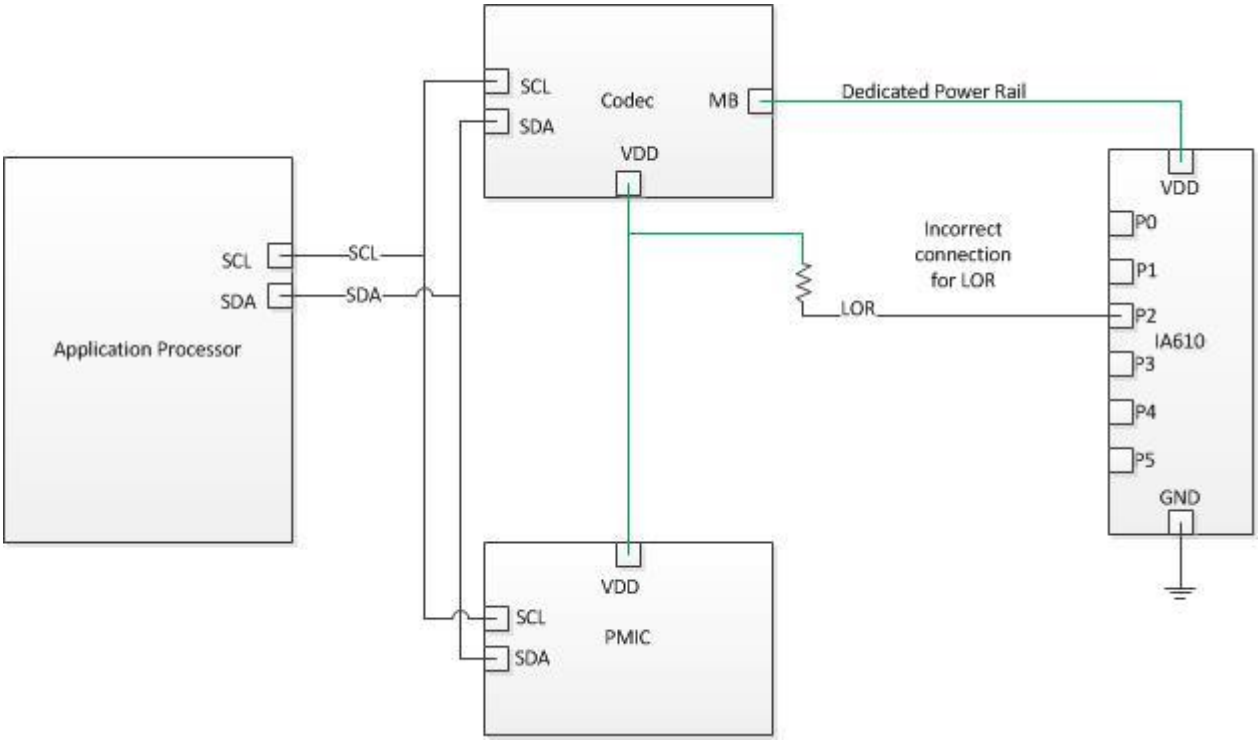

Figure 9 Latch On Reset Incorrect Connection

2. Power Selection

The power source of the IA61x is normally either the microphone bias supply from a Codec, or a rail from a PMIC/regulator. Note the following power source criteria for the IA61x.

1.The power rail can be enabled without toggling any pin on the IA61x.

2.The power rail can be left on in all use cases, including the low-power state.

A main trade off is cost vs complexity. The microphone bias is often more cost effective, while the dedicated power supply is easier to implement.

2.1 Dedicated Power Supply

A *dedicated* power supply covers both a discrete power supply for the IA61x, and a rail from a PMIC in the system. This is often a costlier implementation, requiring an extra rail on a PMIC or dedicated components to generate the power for the IA61x.

The first criterion is satisfied as long as the control interface is isolated from the IA61x. Use a dedicated regulator to meet this condition. With a PMIC, this is only true if the control interface is not shared between the PMIC and the IA61x – similar to Figure 3.

The second criterion is resolved with this approach since the rail can be left on. With a dedicated regulator, once the power supply is enabled, there is no reason to disable the rail. This favors putting the IA61x on a rail that is powered at the start (like VDD_IO) and left on.

With a PMIC, the operating system must be configured to keep this rail enabled at all times. The rail also must be enabled before any transactions on the IA61x control bus occurs.

3. Power Sequencing

The IA61x has an initialization sequence. After it has been initialized the power to the IA61x must be maintained until the device is powered off.

3.1 Initialization Sequence

The power-up sequencing assumes that all pins (VDD_IO, P0-P5) start low. Any pull-up resistors for latch-on reset or I^2C control must be connected to the VDD_IO rail.

The IA61x must receive power on VDD_IO before any pins (P0-P5) toggle. If a pin toggles before VDD_IO is enabled, the IA61x can enter an unknown state. To correct this, have all pins logic low until the VDD_IO voltage level is above 1.7 V. This allows the IA61x to power off all internal components and the IA61x to be in a known state. The time is strongly system dependent and must be evaluated for each design.

Once the VDD_IO reaches the power good state, the bus must remain idle for 12 ms. During this time, the LOR pins are latched into the IA61x, and the IA61x initializes the internal memory and peripherals. Any pin transitions during this time can cause the IA61x to have a different configuration (LOR or pin definition).

After the idle time, the boot sequence or audio pass-through mode can begin. The IA61x has an auto-detect mechanism to determine which configuration it is in. The IA61x API Guide describes the auto-detect mechanism in more detail.

3.2 Deep-Sleep Mode

The IA61x has a deep-sleep mode for when the IA61x is not being used. The system should place the IA61x into this mode, rather than disable power to the IA61x.

3.3 Error Handling

The IA61x has no external RESET to handle unexpected failures. Thus, the IA61x must be power cycled or be self-monitoring.

3.4 Power Cycling

Power cycling the IA61x as a recovery mechanism has the same requirements as the initial power sequencing. The main concern is allowing the power to dissipate in the IA61x before

enabling power on the IA61x again. This is system-dependent and each design must be evaluated to determine the appropriate amount of time that all pins are logic low.

3.5 Self-Monitoring

The IA61x has a watchdog timer to monitor behavior internally. This timer resets the IA61x if the internal code handles an unexpected code path.

The host must be aware of this and expect the IA61x to enter the boot loader at any time. While this is a rare event, it is unpredictable and must be properly guarded for in the application software controlling the IA61x.

Revision History

Revision	Description	Date
1.0	Initial release.	3/26/2018

Model/Reference Number:
Application Note SHP-AN04 Rev 1.0

Phone: (630) 250-5100
Fax: (630) 250-0575
sales@knowles.com

Knowles
1151 Maplewood Drive
Itasca, Illinois 60143