# The LM49370 PCM/I²S Bridge

National Semiconductor
Application Note 1591
Alvin Fok
July 2007

## PCM/I²S Bridge Overview

With an ever increasing feature set found in today's mobile phones, reducing software overhead and baseband processor loading is high priority for any mobile phone design. The most straightforward way to accomplish this without sacrificing any features is to delegate some of the baseband processor's duties over to an external IC.

One such duty is the management of multiple digital audio formats such as mono PCM (for voice) and stereo I²S (for music). Bluetooth headset support and MP3/MP4 playback capability are standard features that facilitate the need to manage different digital audio interfaces within the same system. As shown in Figure 1, a Bluetooth transceiver typically uses a mono PCM audio format during a voice communication whereas an external audio DAC uses a stereo I²S audio format for MP3/MP4 playback.

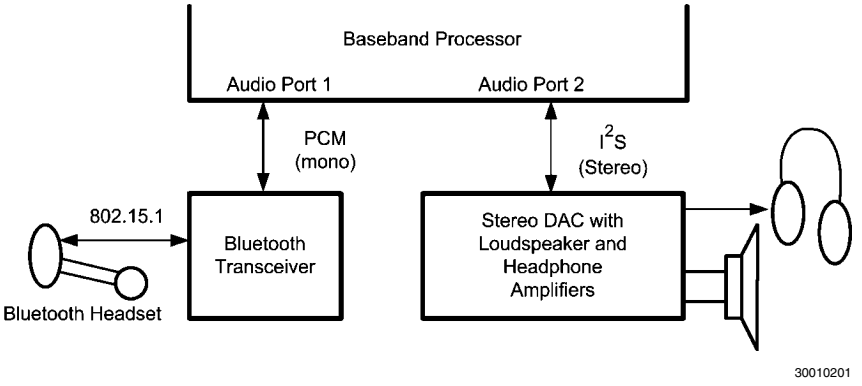


**FIGURE 1. System Level Diagram**

The onboard PCM/I²S bridge of the LM49370 allows the baseband processor to manage just a single digital audio interface thereby reducing processor load and associating software overhead. The LM49370 links directly to the Bluetooth transceiver, freeing up a valuable audio port on the baseband microprocessor. The PCM/I²S bridge can convert a mono PCM interface into a stereo I²S interface or vice versa.

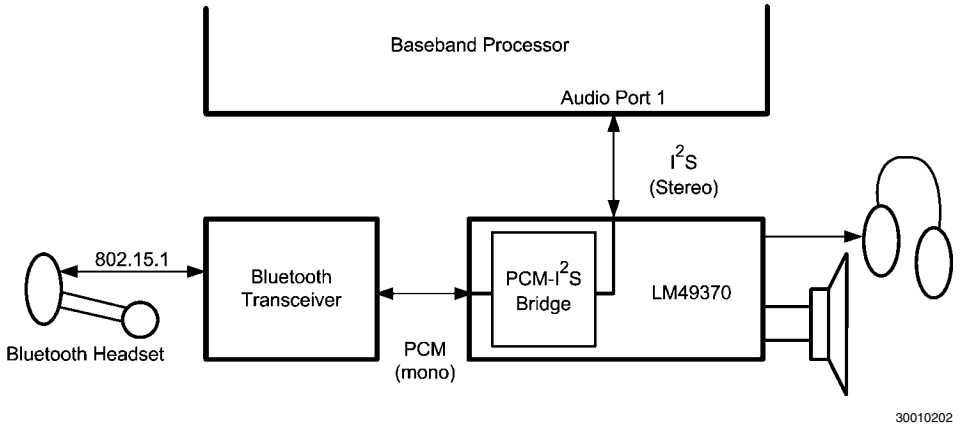


**FIGURE 2. System Level Diagram with LM49370 PCM-I²S Bridge**





 

## PCM/I²S Bridge Explained

The PCM/I²S bridge is specifically targeted for mobile phone applications that require the use of a mono Bluetooth headset. A mono Bluetooth headset is used for voice record, music playback, and two-way voice communication.

In the voice record scenario, the PCM/I²S bridge accepts voice data in mono PCM format at a particular sample rate from a Bluetooth transceiver. The incoming PCM audio data is then converted to a stereo I²S format at a programmable sample rate that is sent out through the LM49370's I²S output port over to the baseband processor for recording.

In the music playback scenario, the PCM/I²S bridge accepts music data in stereo I²S format at a particular sample rate from the baseband processor. The left and right I²S data can then be summed together and converted to a mono PCM format. The converted mono PCM music data is then routed to the Bluetooth transceiver at a programmable sample rate via the LM49370's PCM output port.

For two-way voice communication, it is assumed that the I²S port on the baseband processor handles the voice receive and transmit for this particular example. Also, the I²S and PCM interfaces must run at the same sample rate. The PCM/I²S bridge accepts voice data in mono PCM format at a particular sample rate from a Bluetooth transceiver. The incoming PCM audio data is then converted to a stereo I²S format and is sent out through the LM49370's I²S output port over to the baseband processor for transmit. The PCM/I²S bridge accepts voice data in stereo I²S format from the baseband processor. The left and right I²S data can then be summed together and converted to a mono PCM format. The converted mono PCM music data is then routed to the Bluetooth transceiver via the LM49370's PCM output port for voice receive. For systems that use a dedicated RF IC for voice receive and transmit, the LM49370 features a differential analog input (CP_IN) and output (CP_OUT) pair that directly connect to the analog interface of the RF IC while the LM49370's PCM port still directly interfaces with the Bluetooth transceiver. This an alternative to using the PCM/I²S bridge for two way voice communication via a Bluetooth headset (see Fig. 12).

One limitation of the PCM/I²S Bridge is that both the PCM and I²S interfaces of the LM49370 should not simultaneously operate as slaves. Both the PCM and I²S interfaces can run as masters, or one master and the other slave. However, it is not recommended to run both interfaces as slaves if the PCM and I²S clock sources cannot be synchronized exactly. When using the PCM/I²S Bridge, set the FORCERQ bit to '1' to activate the LM49370's R and Q dividers (see Figure 9) which allow PCM or I²S clock generation in master mode.

The compression format of the PCM interface is transparent to the PCM/I²S bridge. The PCM interface can operate in linear mode, A-law compressed, or μ-law compressed without any impact on PCM/I²S bridge operation. The I²S bus format (standard or left justified) is also transparent to the PCM/I²S bridge.

The PCM/I²S bridge operates with minimal power consumption and also supports sample rate conversion (SRC). However, there are three primary modes of PCM/I²S Bridge operation which are dependent on the sample rates of the PCM and I²S buses. To better understand the different models of operation, two sample rate "families" have been identified, 48kHz and 44.1kHz. The 48kHz sample rate family consists of 8kHz, 16kHz, 24kHz, 32kHz, and 48kHz. The 44.1kHz sample rate family consists of 11.025kHz, 22.05kHz, and 44.1kHz.

The three modes of operation of the PCM/I²S Bridge are:

1) The PCM and I²S interfaces run at the same sample rate without sample rate conversion

2) The PCM and I²S interfaces run at different sample rates that are within the same sample rate family.

3) The PCM and I²S interfaces run at different sample rates that are not of the same family.

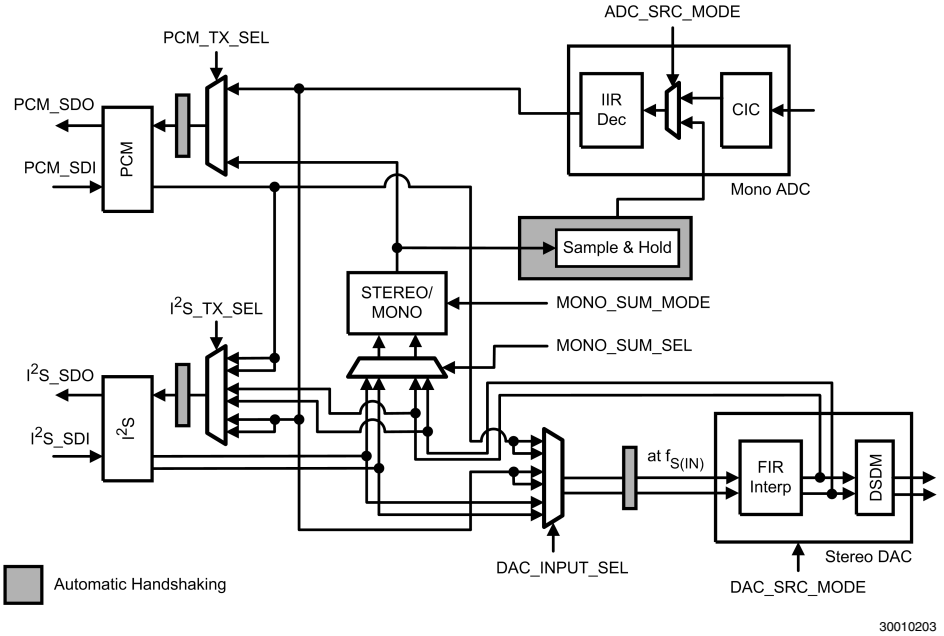


**FIGURE 3. Overall PCM/I²S Bridge Diagram**

## PCM and I²S Running at the Same Sample Rate

The recommended PCM/I²S Bridge mode of operation is to have both the PCM and I²S interfaces run at the same sample rate. This mode of operation consumes the least amount of power because there is no sample rate conversion. The LM49370's stereo DAC and mono ADC can both be completely powered off in this mode of operation. For voice recording through the Bluetooth headset, set I2S_TX_SELECT to '01' (PCM Receiver) as this will route the mono PCM data received from PCM_SDI to I2S_SDO.

The mono signal is automatically copied onto both the left and right channels of the I²S bus. For MP3/MP4 playback (or voice receive via the baseband processor) through the Bluetooth headset, set PCM_TX_SELECT to '1' (Mono Sum Circuit) as this will route the stereo I²S data received from I2S_SDI to PCM_SDO via the mono sum circuit. The mono sum circuit acts like a stereo to mono converter. It takes the stereo I²S data and produces a mono output. The mono output can be either the I²S left channel, the I²S right channel, or a mono signal created by adding the left and right I²S channels. This is controlled by MONO_SUM_MODE.

This is also the only PCM/I²S Bridge mode of operation that allows two-way voice communication via Bluetooth headset for systems that implement the baseband processor's I²S port for voice receive and transmit.

**TABLE 1. PCM/I²S Bridge Settings (I²S and PCM Operating at the Same Sample Rate)**

| I²C Field (Note 1) | Setting | Result |
|---|---|---|
| PCM_TX_SEL | 1 | Mono Sum Circuit |
| I2S_TX_SEL | 01 | PCM Receiver |
| DAC_INPUT_SEL | 11 | Disabled |
| MONO_SUM_SEL | 1 | I²S Receiver |
| MONO_SUM_MODE | 00 | Mono Out = (Left I²S + Right I²S) / 2 |
| ADC_SRC_MODE | 0 | Disabled |
| DAC_SRC_MODE | 0 | Disabled |

**Note 1:** Please refer to the LM49370 datasheet for detailed information regarding the I²C register settings.

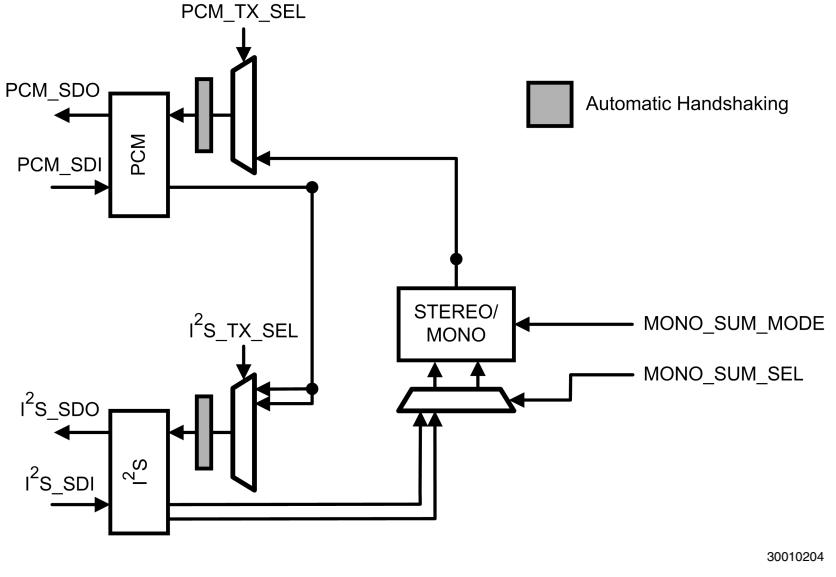


**FIGURE 4. PCM and I²S Running at the Same Sample Rate (Voice Communication, Voice Record, or MP3/MP4 Playback)**

## I²S and PCM Running at Different Sample Rates Within the Same Sample Rate Family

In the case of system level restraints, the I²S and PCM interfaces may have to operate at different sample rates while running simultaneously. The PCM/I²S Bridge can be used as a sample rate converter either to upsample or downsample the incoming data stream. As long as the sample rates of the I²S and PCM interfaces are within the same sample rate family, portions of the LM49370's stereo DAC and mono ADC can be powered down.

This mode of operation consumes more power due to the sample rate conversion process. Therefore, it is recommended to avoid sample rate conversion if it is not required.

This mode requires access to the stereo DAC's interpolated output. The interpolated DAC output can be resampled directly through the I²S interface or it can be sent to the mono sum circuit and then fed to the Sample and Hold block. The output of the Sample and Hold block is then decimated by the ADC's IIR filter. The resulting ADC output can then be routed to either the I²S or PCM interface.

Throughout this document, there will be many references to $f_{S(IN)}$ and $f_{S(OUT)}$. The variable $f_{S(IN)}$ represents the sample rate of the receiving interface (I²S or PCM) of the PCM/I²S Bridge, and $f_{S(OUT)}$ represents the sample rate of the transmitting interface (I²S or PCM) of the PCM/I²S Bridge. For Bluetooth voice recording, $f_{S(IN)}$ represents the PCM sample rate and $f_{S(OUT)}$ represents the I²S sample rate. For Bluetooth MP3/MP4 playback, $f_{S(IN)}$ represents the I²S sample rate and $f_{S(OUT)}$ represents the PCM sample rate.

The I²S (or PCM) interface can directly resample the DAC's interpolated output only if the following conditions are met:

**Condition 1: $f_{S(IN)} < f_{S(OUT)}$ , which is equivalent to upsampling**

**Condition 2: $((f_{S(IN)} / f_{S(OUT)}) * (OSR))$ = an integer number, where OSR is the oversampling ratio of the DAC and ADC.**

If these conditions are not met, then the interpolated DAC output has to be resampled using the Sample and Hold block. One such scenario that requires the use of the Sample and Hold block is converting an $f_{S(IN)}$ of 8kHz to an $f_{S(OUT)}$ of 24kHz. Assuming an OSR of 128, Condition 2 is violated because ((8kHz) / 24kHz) * (128)) results in a non-integer value of 42.667. Another scenario that requires the use of the Sample and Hold block is downsampling an $f_{S(IN)}$ of 48kHz to an $f_{S(OUT)}$ of 8kHz, which is violation of Condition 1. It is important to note that the DAC and ADC can operate with an OSR of 125 or 128.

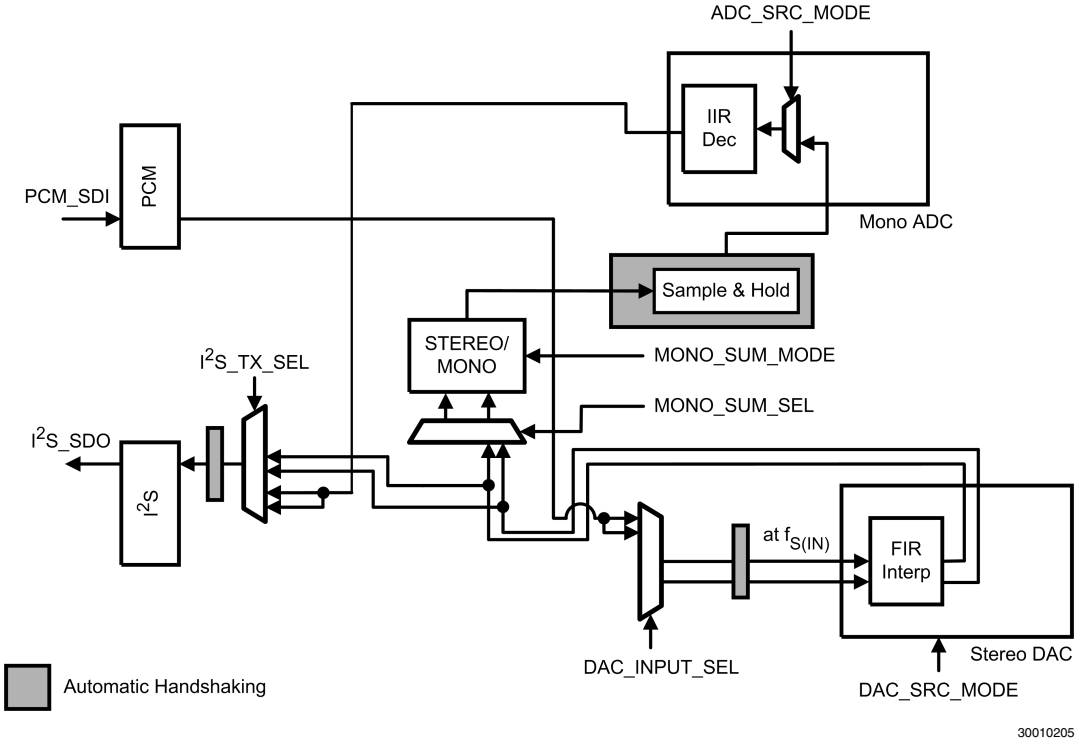


**FIGURE 5. PCM and I²S Running at Different Sample Rates Within the Same Sample Rate Family (Voice Record)**

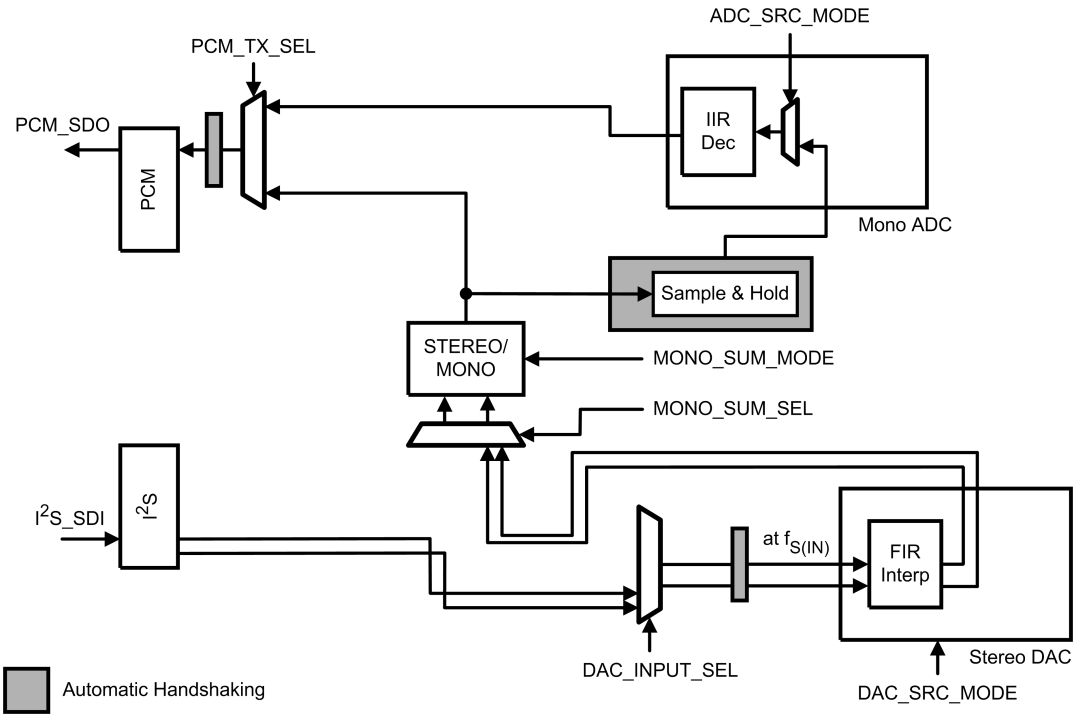


**FIGURE 6. PCM and I²S Running at Different Sample Rates Within the Same Sample Rate Family (MP3/MP4 Playback)**

The following tables can be used as a guide to correctly set the PCM/I²S Bridge where $f_{S(IN)}$ and $f_{S(OUT)}$ are from the same sample rate family:

**TABLE 2. Converting 8kHz to Other Sample Rates Within the Same Family (OSR = 128)**

| $f_{S(IN)}$ | $f_{S(OUT)}$ | Voice Record Settings | MP3 Playback Settings |
|---|---|---|---|
| 8kHz | 8kHz | Refer to Table 1 | Refer to Table 1 |
| | 16kHz | PCM_TX_SELECT = X | PCM_TX_SELECT = 1 |
| | | I2S_TX_SELECT = 10 | I2S_TX_SELECT = XX |
| | | DAC_INPUT_SEL = 01 | DAC_INPUT_SEL = 00 |
| | | MONO_SUM_SEL = 0 | MONO_SUM_SEL = 0 |
| | | MONO_SUM_MODE = 00 | MONO_SUM_MODE = 00 |
| | | ADC_SRC_MODE = 0 | ADC_SRC_MODE = 0 |
| | | DAC_SRC_MODE = 1 | DAC_SRC_MODE = 1 |
| | 24kHz | PCM_TX_SELECT = X | PCM_TX_SELECT = 0 |
| | | I2S_TX_SELECT = 00 | I2S_TX_SELECT = XX |
| | | DAC_INPUT_SEL = 01 | DAC_INPUT_SEL = 00 |
| | | MONO_SUM_SEL = 0 | MONO_SUM_SEL = 0 |
| | | MONO_SUM_MODE = 00 | MONO_SUM_MODE = 00 |
| | | ADC_SRC_MODE = 1 | ADC_SRC_MODE = 1 |
| | | DAC_SRC_MODE = 1 | DAC_SRC_MODE = 1 |
| | 32kHz | PCM_TX_SELECT = X | PCM_TX_SELECT = 1 |
| | | I2S_TX_SELECT = 10 | I2S_TX_SELECT = XX |
| | | DAC_INPUT_SEL = 01 | DAC_INPUT_SEL = 00 |
| | | MONO_SUM_SEL = 0 | MONO_SUM_SEL = 0 |
| | | MONO_SUM_MODE = 00 | MONO_SUM_MODE = 00 |
| | | ADC_SRC_MODE = 0 | ADC_SRC_MODE = 0 |
| | | DAC_SRC_MODE = 1 | DAC_SRC_MODE = 1 |
| | 48kHz | PCM_TX_SELECT = X | PCM_TX_SELECT = 0 |
| | | I2S_TX_SELECT = 00 | I2S_TX_SELECT = XX |
| | | DAC_INPUT_SEL = 01 | DAC_INPUT_SEL = 00 |
| | | MONO_SUM_SEL = 0 | MONO_SUM_SEL = 0 |
| | | MONO_SUM_MODE = 00 | MONO_SUM_MODE = 00 |
| | | ADC_SRC_MODE = 1 | ADC_SRC_MODE = 1 |
| | | DAC_SRC_MODE = 1 | DAC_SRC_MODE = 1 |

**TABLE 3. Converting 16kHz to Other Sample Rates Within the Same Family (OSR = 128)**

| $f_{S(IN)}$ | $f_{S(OUT)}$ | Voice Record Settings | MP3 Playback Settings |
|---|---|---|---|
| 16kHz | 8kHz | PCM_TX_SELECT = X | PCM_TX_SELECT = 0 |
| | | I2S_TX_SELECT = 00 | I2S_TX_SELECT = XX |
| | | DAC_INPUT_SEL = 01 | DAC_INPUT_SEL = 00 |
| | | MONO_SUM_SEL = 0 | MONO_SUM_SEL = 0 |
| | | MONO_SUM_MODE = 00 | MONO_SUM_MODE = 00 |
| | | ADC_SRC_MODE = 1 | ADC_SRC_MODE = 1 |
| | | DAC_SRC_MODE = 1 | DAC_SRC_MODE = 1 |
| | 16kHz | Refer to Table 1 | Refer to Table 1 |
| | 24kHz | PCM_TX_SELECT = X | PCM_TX_SELECT = 0 |
| | | I2S_TX_SELECT = 00 | I2S_TX_SELECT = XX |
| | | DAC_INPUT_SEL = 01 | DAC_INPUT_SEL = 00 |
| | | MONO_SUM_SEL = 0 | MONO_SUM_SEL = 0 |
| | | MONO_SUM_MODE = 00 | MONO_SUM_MODE = 00 |
| | | ADC_SRC_MODE = 1 | ADC_SRC_MODE = 1 |
| | | DAC_SRC_MODE = 1 | DAC_SRC_MODE = 1 |
| | 32kHz | PCM_TX_SELECT = X | PCM_TX_SELECT = 1 |
| | | I2S_TX_SELECT = 10 | I2S_TX_SELECT = XX |
| | | DAC_INPUT_SEL = 01 | DAC_INPUT_SEL = 00 |
| | | MONO_SUM_SEL = 0 | MONO_SUM_SEL = 0 |
| | | MONO_SUM_MODE = 00 | MONO_SUM_MODE = 00 |
| | | ADC_SRC_MODE = 0 | ADC_SRC_MODE = 0 |
| | | DAC_SRC_MODE = 1 | DAC_SRC_MODE = 1 |
| | 48kHz | PCM_TX_SELECT = X | PCM_TX_SELECT = 0 |
| | | I2S_TX_SELECT = 00 | I2S_TX_SELECT = XX |
| | | DAC_INPUT_SEL = 01 | DAC_INPUT_SEL = 00 |
| | | MONO_SUM_SEL = 0 | MONO_SUM_SEL = 0 |
| | | MONO_SUM_MODE = 00 | MONO_SUM_MODE = 00 |
| | | ADC_SRC_MODE = 1 | ADC_SRC_MODE = 1 |
| | | DAC_SRC_MODE = 1 | DAC_SRC_MODE = 1 |

**TABLE 4. Converting 24kHz to Other Sample Rates Within the Same Family (OSR = 128)**

| $f_{S(IN)}$ | $f_{S(OUT)}$ | Voice Record Settings | MP3 Playback Settings |
|---|---|---|---|
| 24kHz | 8kHz | PCM_TX_SELECT = X | PCM_TX_SELECT = 0 |
| | | I2S_TX_SELECT = 00 | I2S_TX_SELECT = XX |
| | | DAC_INPUT_SEL = 01 | DAC_INPUT_SEL = 00 |
| | | MONO_SUM_SEL = 0 | MONO_SUM_SEL = 0 |
| | | MONO_SUM_MODE = 00 | MONO_SUM_MODE = 00 |
| | | ADC_SRC_MODE = 1 | ADC_SRC_MODE = 1 |
| | | DAC_SRC_MODE = 1 | DAC_SRC_MODE = 1 |
| | 16kHz | PCM_TX_SELECT = X | PCM_TX_SELECT = 0 |
| | | I2S_TX_SELECT = 00 | I2S_TX_SELECT = XX |
| | | DAC_INPUT_SEL = 01 | DAC_INPUT_SEL = 00 |
| | | MONO_SUM_SEL = 0 | MONO_SUM_SEL = 0 |
| | | MONO_SUM_MODE = 00 | MONO_SUM_MODE = 00 |
| | | ADC_SRC_MODE = 1 | ADC_SRC_MODE = 1 |
| | | DAC_SRC_MODE = 1 | DAC_SRC_MODE = 1 |
| | 24kHz | Refer to Table 1 | Refer to Table 1 |
| | 32kHz | PCM_TX_SELECT = X | PCM_TX_SELECT = 1 |
| | | I2S_TX_SELECT = 10 | I2S_TX_SELECT = XX |
| | | DAC_INPUT_SEL = 01 | DAC_INPUT_SEL = 00 |
| | | MONO_SUM_SEL = 0 | MONO_SUM_SEL = 0 |
| | | MONO_SUM_MODE = 00 | MONO_SUM_MODE = 00 |
| | | ADC_SRC_MODE = 0 | ADC_SRC_MODE = 0 |
| | | DAC_SRC_MODE = 1 | DAC_SRC_MODE = 1 |
| | 48kHz | PCM_TX_SELECT = X | PCM_TX_SELECT = 1 |
| | | I2S_TX_SELECT = 10 | I2S_TX_SELECT = XX |
| | | DAC_INPUT_SEL = 01 | DAC_INPUT_SEL = 00 |
| | | MONO_SUM_SEL = 0 | MONO_SUM_SEL = 0 |
| | | MONO_SUM_MODE = 00 | MONO_SUM_MODE = 00 |
| | | ADC_SRC_MODE = 0 | ADC_SRC_MODE = 0 |
| | | DAC_SRC_MODE = 1 | DAC_SRC_MODE = 1 |

**TABLE 5. Converting 32kHz to Other Sample Rates Within the Same Family (OSR = 128)**

| $f_{S(IN)}$ | $f_{S(OUT)}$ | Voice Record Settings | MP3 Playback Settings |
|---|---|---|---|
| 32kHz | 8kHz | PCM_TX_SELECT = X | PCM_TX_SELECT = 0 |
| | | I2S_TX_SELECT = 00 | I2S_TX_SELECT = XX |
| | | DAC_INPUT_SEL = 01 | DAC_INPUT_SEL = 00 |
| | | MONO_SUM_SEL = 0 | MONO_SUM_SEL = 0 |
| | | MONO_SUM_MODE = 00 | MONO_SUM_MODE = 00 |
| | | ADC_SRC_MODE = 1 | ADC_SRC_MODE = 1 |
| | | DAC_SRC_MODE = 1 | DAC_SRC_MODE = 1 |
| | 16kHz | PCM_TX_SELECT = X | PCM_TX_SELECT = 0 |
| | | I2S_TX_SELECT = 00 | I2S_TX_SELECT = XX |
| | | DAC_INPUT_SEL = 01 | DAC_INPUT_SEL = 00 |
| | | MONO_SUM_SEL = 0 | MONO_SUM_SEL = 0 |
| | | MONO_SUM_MODE = 00 | MONO_SUM_MODE = 00 |
| | | ADC_SRC_MODE = 1 | ADC_SRC_MODE = 1 |
| | | DAC_SRC_MODE = 1 | DAC_SRC_MODE = 1 |
| | 24kHz | PCM_TX_SELECT = X | PCM_TX_SELECT = 0 |
| | | I2S_TX_SELECT = 00 | I2S_TX_SELECT = XX |
| | | DAC_INPUT_SEL = 01 | DAC_INPUT_SEL = 00 |
| | | MONO_SUM_SEL = 0 | MONO_SUM_SEL = 0 |
| | | MONO_SUM_MODE = 00 | MONO_SUM_MODE = 00 |
| | | ADC_SRC_MODE = 1 | ADC_SRC_MODE = 1 |
| | | DAC_SRC_MODE = 1 | DAC_SRC_MODE = 1 |
| | 32kHz | Refer to Table 1 | Refer to Table 1 |
| | 48kHz | PCM_TX_SELECT = X | PCM_TX_SELECT = 0 |
| | | I2S_TX_SELECT = 00 | I2S_TX_SELECT = XX |
| | | DAC_INPUT_SEL = 01 | DAC_INPUT_SEL = 00 |
| | | MONO_SUM_SEL = 0 | MONO_SUM_SEL = 0 |
| | | MONO_SUM_MODE = 00 | MONO_SUM_MODE = 00 |
| | | ADC_SRC_MODE = 1 | ADC_SRC_MODE = 1 |
| | | DAC_SRC_MODE = 1 | DAC_SRC_MODE = 1 |

**TABLE 6. Converting 48kHz to Other Sample Rates Within the Same Family (OSR = 128)**

| $f_{S(IN)}$ | $f_{S(OUT)}$ | Voice Record Settings | MP3 Playback Settings |
|---|---|---|---|
| 48kHz | 8kHz | PCM_TX_SELECT = X | PCM_TX_SELECT = 0 |
| | | I2S_TX_SELECT = 00 | I2S_TX_SELECT = XX |
| | | DAC_INPUT_SEL = 01 | DAC_INPUT_SEL = 00 |
| | | MONO_SUM_SEL = 0 | MONO_SUM_SEL = 0 |
| | | MONO_SUM_MODE = 00 | MONO_SUM_MODE = 00 |
| | | ADC_SRC_MODE = 1 | ADC_SRC_MODE = 1 |
| | | DAC_SRC_MODE = 1 | DAC_SRC_MODE = 1 |
| | 16kHz | PCM_TX_SELECT = X | PCM_TX_SELECT = 0 |
| | | I2S_TX_SELECT = 00 | I2S_TX_SELECT = XX |
| | | DAC_INPUT_SEL = 01 | DAC_INPUT_SEL = 00 |
| | | MONO_SUM_SEL = 0 | MONO_SUM_SEL = 0 |
| | | MONO_SUM_MODE = 00 | MONO_SUM_MODE = 00 |
| | | ADC_SRC_MODE = 1 | ADC_SRC_MODE = 1 |
| | | DAC_SRC_MODE = 1 | DAC_SRC_MODE = 1 |
| | 24kHz | PCM_TX_SELECT = X | PCM_TX_SELECT = 0 |
| | | I2S_TX_SELECT = 00 | I2S_TX_SELECT = XX |
| | | DAC_INPUT_SEL = 01 | DAC_INPUT_SEL = 00 |
| | | MONO_SUM_SEL = 0 | MONO_SUM_SEL = 0 |
| | | MONO_SUM_MODE = 00 | MONO_SUM_MODE = 00 |
| | | ADC_SRC_MODE = 1 | ADC_SRC_MODE = 1 |
| | | DAC_SRC_MODE = 1 | DAC_SRC_MODE = 1 |
| | 32kHz | PCM_TX_SELECT = X | PCM_TX_SELECT = 0 |
| | | I2S_TX_SELECT = 00 | I2S_TX_SELECT = XX |
| | | DAC_INPUT_SEL = 01 | DAC_INPUT_SEL = 00 |
| | | MONO_SUM_SEL = 0 | MONO_SUM_SEL = 0 |
| | | MONO_SUM_MODE = 00 | MONO_SUM_MODE = 00 |
| | | ADC_SRC_MODE = 1 | ADC_SRC_MODE = 1 |
| | | DAC_SRC_MODE = 1 | DAC_SRC_MODE = 1 |
| | 48kHz | Refer to Table 1 | Refer to Table 1 |

**TABLE 7. Converting any 48kHz Related Sample Rate to Other Sample Rates Within the Same Family (OSR = 125)**

| $f_{S(IN)}$ | $f_{S(OUT)}$ | Voice Record Settings | MP3 Playback Settings |
|---|---|---|---|
| Any 48kHz Related Sample Rate | Any 48kHz Related Sample Rate that is not $f_{S(IN)}$ | PCM_TX_SELECT = X | PCM_TX_SELECT = 0 |
| | | I2S_TX_SELECT = 00 | I2S_TX_SELECT = XX |
| | | DAC_INPUT_SEL = 01 | DAC_INPUT_SEL = 00 |
| | | MONO_SUM_SEL = 0 | MONO_SUM_SEL = 0 |
| | | MONO_SUM_MODE = 00 | MONO_SUM_MODE = 00 |
| | | ADC_SRC_MODE = 1 | ADC_SRC_MODE = 1 |
| | | DAC_SRC_MODE = 1 | DAC_SRC_MODE = 1 |

Although the tables only cover the 48kHz sample rate family, the same principles can be applied to sample rate conversion within the 44.1kHz sample rate family.

## PCM and I²S Running at Sample Rates from Different Sample Rate Families

Whenever the sample rates of the I²S and PCM interfaces are running at sample rates from different sample rate families (for example $f_{S(IN)}$ = 44.1kHz and $f_{S(OUT)}$ = 8kHz), the PCM/I²S bridge requires that the stereo DAC and mono ADC be fully turned on. The incoming digital audio path (either I²S or PCM) has to be routed to the stereo DAC in order to do the complete digital to analog conversion. The left and right analog DAC outputs are then summed together and routed back through the ADC in order to perform the analog to digital conversion.

Because the entire portion of both the stereo DAC and mono ADC have to be turned on, this particular PCM/I²S bridge mode of operation consumes the most amount of power. This mode of operation is not preferred, but because of system level restrictions this mode of operation may be the only alternative.

There are also limitations on the master clock frequency (MCLK) for this particular mode of operation. The LM49370 has one available PLL. This PLL can be used to generate the required frequencies from one sample rate family. The required clock frequencies from the other sample rate family must be directly divided down from MCLK. If the LM49370's internal clock dividers cannot directly divide down MCLK to the required clock frequencies, the PCM/I²S Bridge cannot be used. Typically, the PLL should handle the 44.1kHz related frequencies while the 48kHz related frequencies should be divided directly from MCLK.

When running the DAC and ADC in 128 OSR mode, an MCLK of 12.288MHz or 11.2896MHz or an integer factor of either clock will guarantee PCM/I²S Bridge operation for handling sampling rates of different families. For 125 OSR mode, and MCLK of 12MHz or 11.025MHz or an integer factor of either clock will guarantee PCM/I²S Bridge operation for handling sample rates of different families. If a 13MHz MCLK or an integer factor of that clock is available, the internal dividers can generate an 8kHz sample rate in 125 OSR mode, while the PLL can handle the 44.1kHz related frequencies.

**TABLE 8. Converting a Sample Rate to Another Sample Rate from a Different Sample Rate Family**

| $f_{S(IN)}$ | $f_{S(OUT)}$ | Voice Record Settings | MP3 Playback Settings |
|---|---|---|---|
| Any 44.1kHz Related Sample Rate | Any 48kHz Related Sample Rate | PCM_TX_SELECT = X | PCM_TX_SELECT = 0 |
| | | I2S_TX_SELECT = 00 | I2S_TX_SELECT = XX |
| | | DAC_INPUT_SEL = 01 | DAC_INPUT_SEL = 00 |
| | | MONO_SUM_SEL = X | MONO_SUM_SEL = X |
| | | MONO_SUM_MODE = XX | MONO_SUM_MODE = XX |
| | | ADC_SRC_MODE = 0 | ADC_SRC_MODE = 0 |
| | | DAC_SRC_MODE = 0 | DAC_SRC_MODE = 0 |
| Any 48kHz Related Sample Rate | Any 44.1kHz Related Sample Rate | PCM_TX_SELECT = X | PCM_TX_SELECT = 0 |
| | | I2S_TX_SELECT = 00 | I2S_TX_SELECT = XX |
| | | DAC_INPUT_SEL = 01 | DAC_INPUT_SEL = 00 |
| | | MONO_SUM_SEL = X | MONO_SUM_SEL = X |
| | | MONO_SUM_MODE = XX | MONO_SUM_MODE = XX |
| | | ADC_SRC_MODE = 0 | ADC_SRC_MODE = 0 |
| | | DAC_SRC_MODE = 0 | DAC_SRC_MODE = 0 |

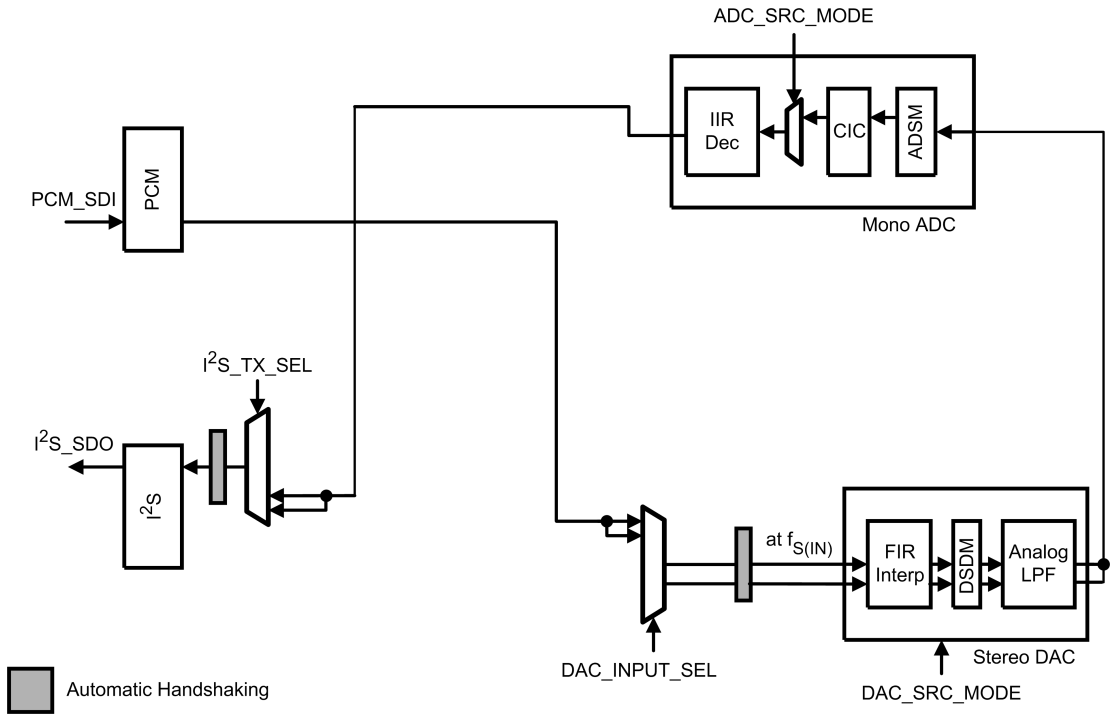


**FIGURE 7. PCM and I²S Running at Different Sample Rates from Different Sample Rate Families (Voice Record)**

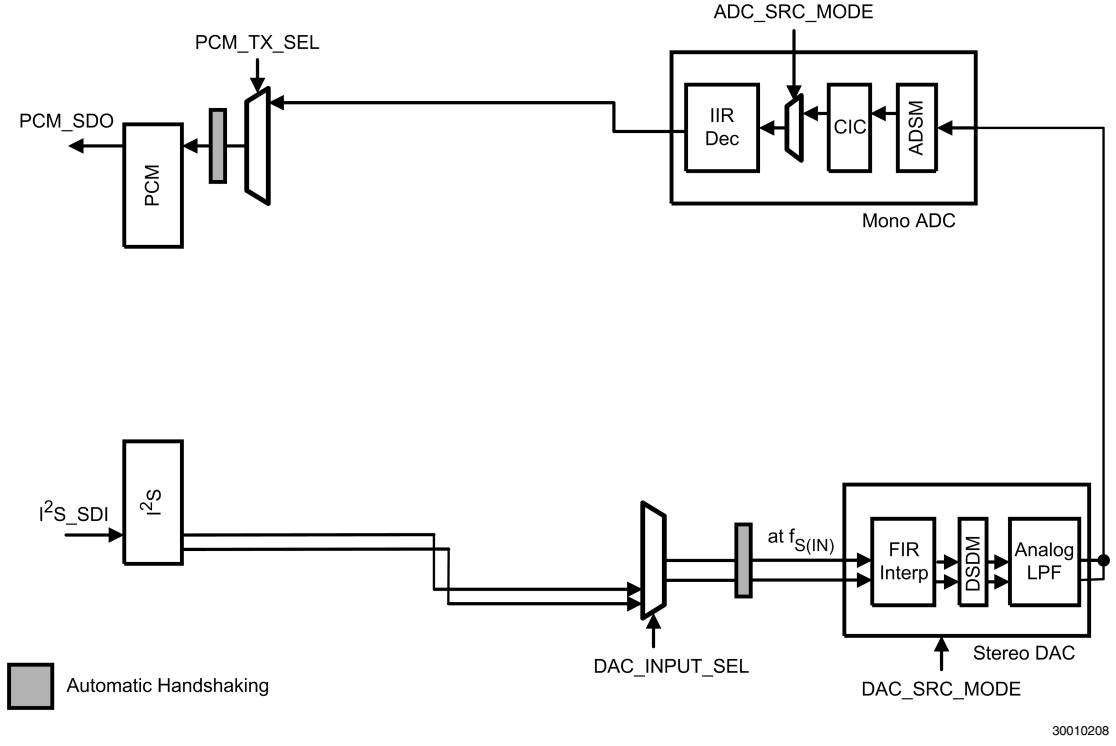


**FIGURE 8. PCM and I²S Running at Different Sample Rates from Different Sample Rate Families (MP3/MP4 Playback)**

## Setting the Correct Clocks

For proper PCM/I²S bridge operation, it is essential to have the correct clocks running on both the PCM and I²S interfaces. The stereo DAC should operate at a clock frequency that is based on the incoming sample rate of the bridge (Equation 1).

$$DAC\_CLOCK = (f_{S(IN)} * OSR) \quad (1)$$

The LM49370's R divider directly divides the incoming MCLK or the PLL output clock to generate DAC_CLOCK. The mono ADC, if used, should operate at a clock frequency that is based on the outgoing sample rate of the bridge (Equation 2).

$$ADC\_CLOCK = (f_{S(OUT)} * OSR) \quad (2)$$

The LM49370's Q divider generates the required ADC_CLOCK.

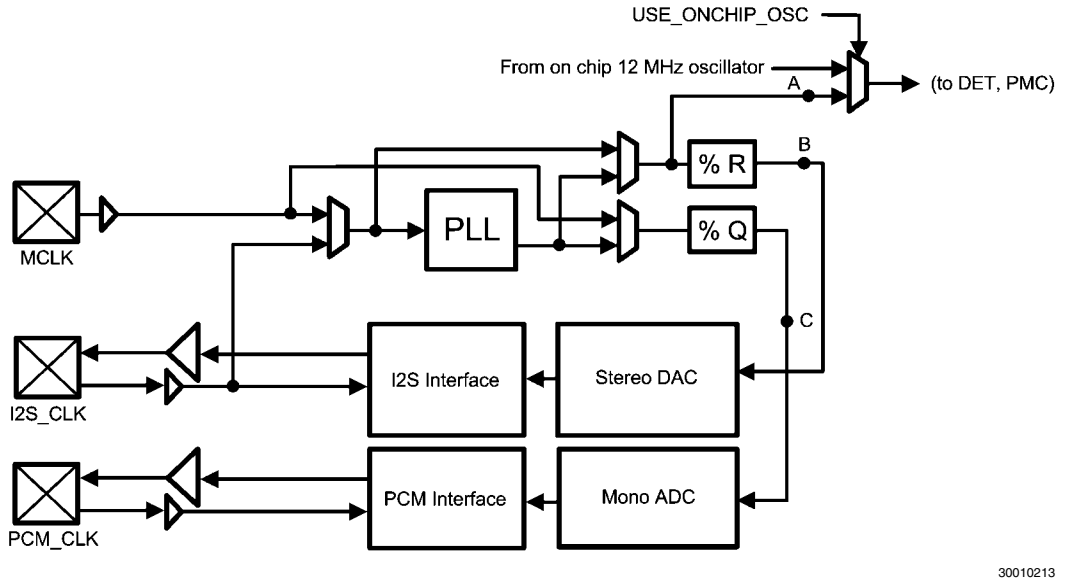


**FIGURE 9. LM49370 Clock Network (Note 2)**

The clock outputs of the R divider (DAC_CLOCK) and the Q divider (ADC_CLOCK) are also used to provide the required clocks for the PCM and I²S interfaces. If the I²S is running in master mode, the I2S_CLOCK_GEN block divides DAC_CLOCK (or ADC_CLOCK) to generate the required I²S clock frequency (I2S_CLOCK). The I2S_WS_GEN block divides I2S_CLOCK to generate the proper I²S sync signal (I2S_WS). The correct frequencies of I2S_WS and I2S_CLOCK are calculated by Equations (3) and (4).

$$I2S\_WS = f_{S(I^2S)} \quad (3)$$

$$I2S\_CLOCK = (\#\ of\ bits\ per\ I^2S\ word) * (2) * f_{S(I^2S)} \quad (4)$$

If the desired I²S sampling rate is 48kHz, then I2S_WS = 48kHz. Assuming an I²S word length of 16 bits, then I2S_CLOCK = 1.536MHz.

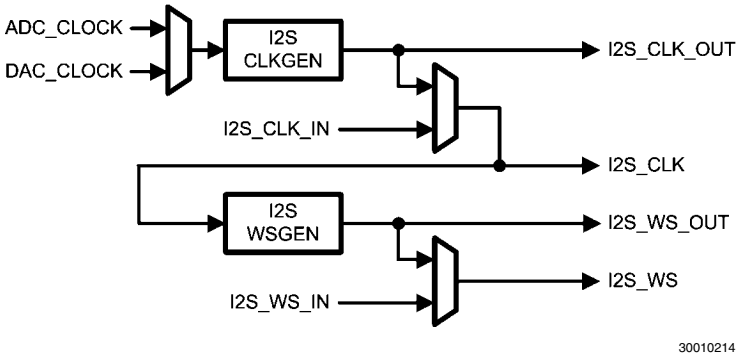


**FIGURE 10. Audio Port CLOCK/SYNC Options**

If the PCM interface is running in master mode, the PCM_CLOCK_GEN block divides DAC_CLOCK (or ADC_CLOCK) to generate the required PCM clock frequency (PCM_CLOCK). The PCM_SYNC_GEN block divides PCM_CLOCK to generate the proper PCM sync signal (PCM_SYNC). The correct frequencies of PCM_SYNC and PCM_CLOCK are calculated by Equations (5) and (6).

$$PCM\_SYNC = f_{S(PCM)} \quad (5)$$

$$PCM\_CLOCK = (\#\ of\ bits\ per\ PCM\ word) * f_{S(PCM)} \quad (6)$$

If the desired PCM sampling rate is 8kHz and the PCM word length is 16 bits, then PCM_SYNC = 8kHz and PCM_CLOCK = 128kHz.

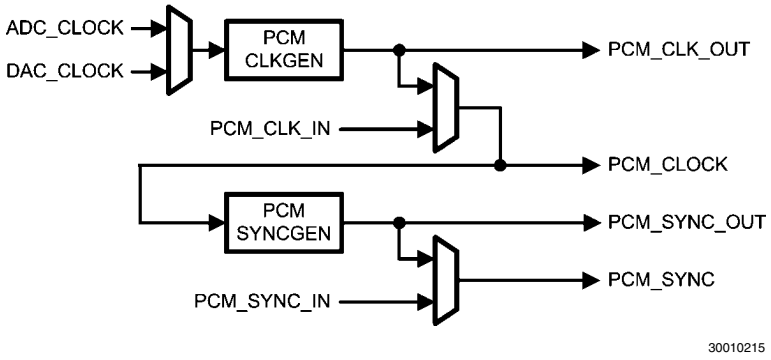


**FIGURE 11. PCM Audio Port CLOCK/SYNC Options**

**Note 2:** Please refer to the LM49370 datasheet on how to set the I²C registers for the PLL, R divider, Q divider, I²S port, and PCM port.

## Two-Way Voice Communication or Music Playback via Bluetooth Headset Without Using the PCM/I²S Bridge

For mobile phone designs that implement a dedicated RF IC (voice modem) for voice communication, the PCM/I²S Bridge can be bypassed as long as the RF IC has an analog interface that can connect directly to the differential inputs (CP_IN) and outputs (CP_OUT) of the LM49370.

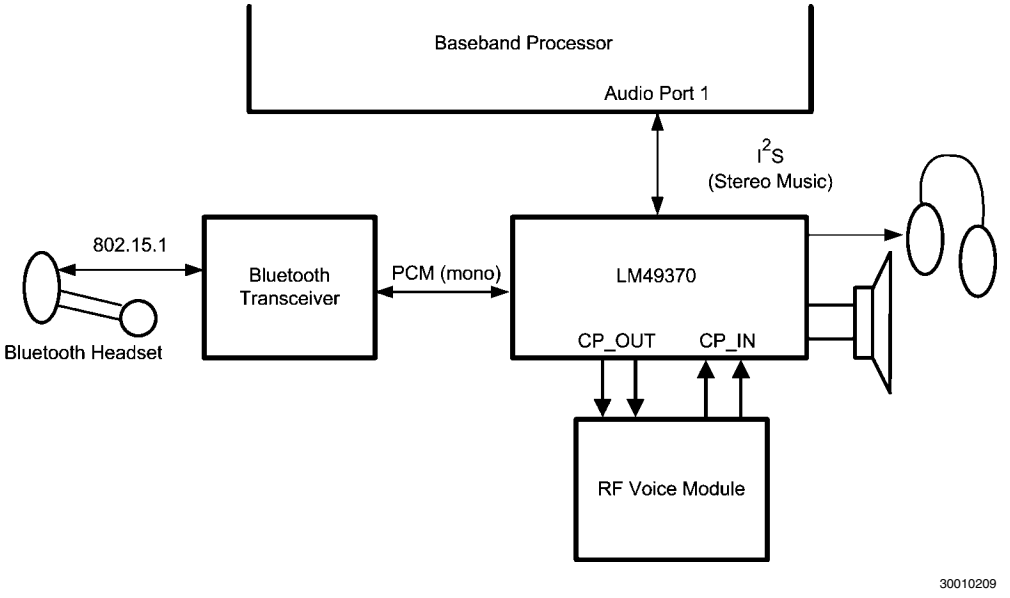


**FIGURE 12. Using the LM49370 with a Dedicated RF Module for Two-way Voice Communication via Bluetooth Headset**

With this type of configuration (Fig 12), the LM49370's I²S port is not needed for two-way voice communication. Incoming analog voice data from the RF voice module is sent from CP_IN to the LM49370's ADC. After the analog to digital conversion, the incoming voice data is routed to the Bluetooth receiver through the PCM output port. Outgoing voice data from the Bluetooth headset is sent to the LM49370's DAC through the PCM input port. After the digital to analog conversion, the outgoing voice data is sent to the RF voice module through the CP_OUT differential output for voice transmit.

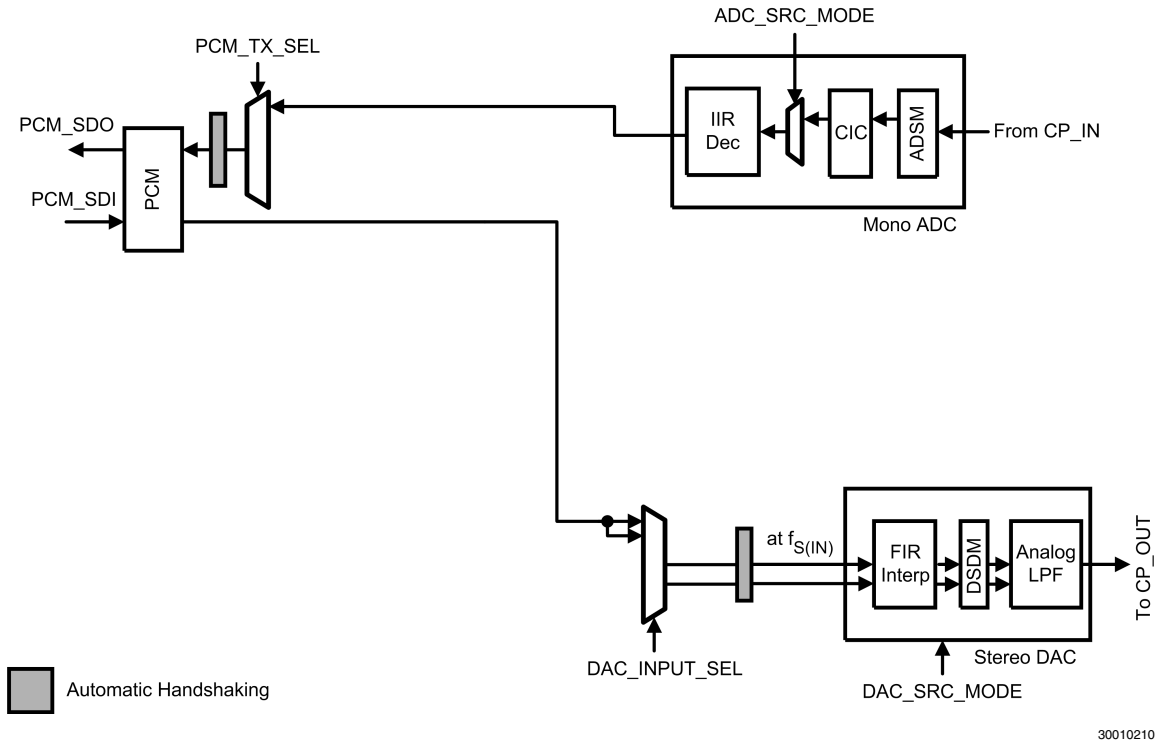


**FIGURE 13. An Alternative Method for Two-way Voice Communication via Bluetooth**

The LM49370 features a set of stereo analog inputs (AUX_L and AUX_R) that can route stereo content from a FM radio module or TV tuner module to the Bluetooth transceiver.

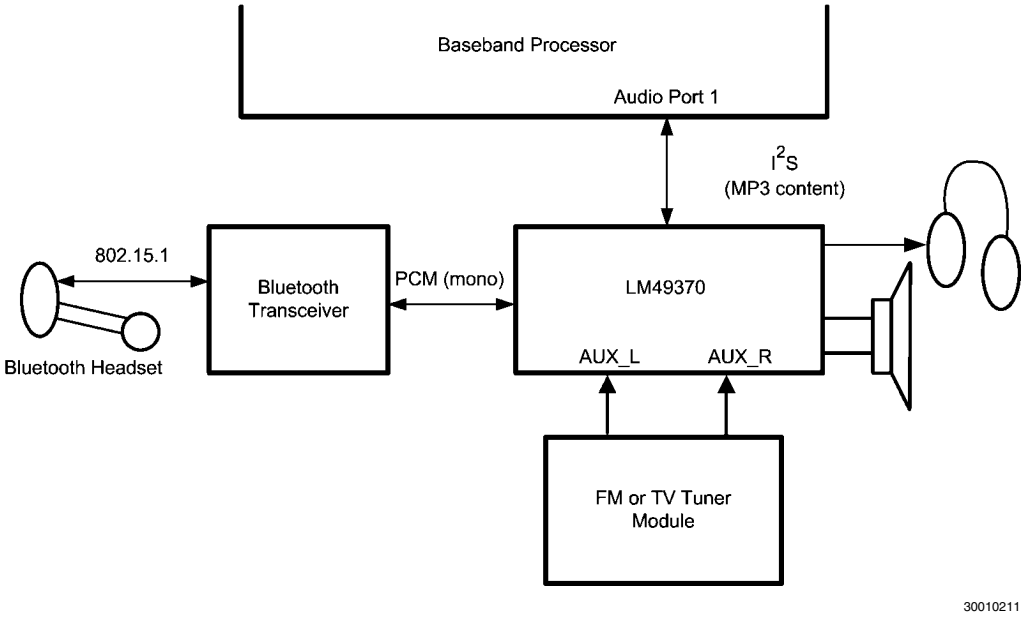


**FIGURE 14. Using the LM49370 to Route FM Radio or TV Audio to a Bluetooth Headset**

The analog audio content received from AUX_L and AUX_R inputs can be summed together to form a mono signal that is then routed to the ADC. After the analog to digital conversion, the mono signal is sent to the Bluetooth transceiver through the PCM output port.

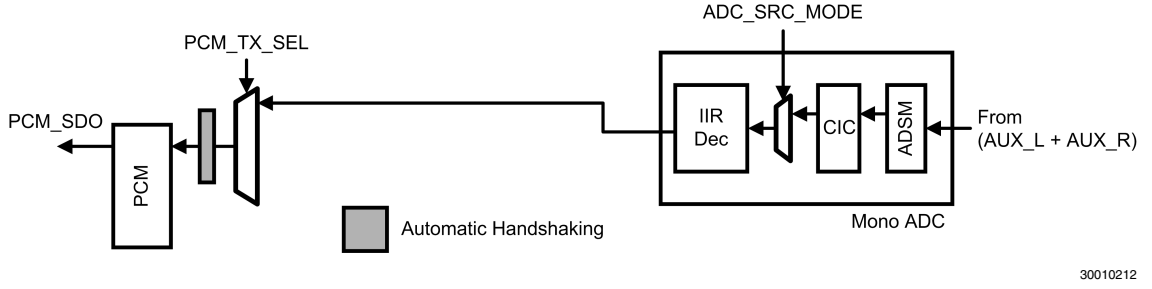


**FIGURE 15. FM Radio or TV Tuner Playback via Bluetooth Headset**

## Stereo Bluetooth Headsets

The LM49370 is primarily targeted for mobile phone scenarios that require the use of a mono Bluetooth headset. For scenarios that require the use of a stereo Bluetooth headset, the Bluetooth transceiver and baseband processor must be A2DP (Advanced Audio Distribution Profile) capable. A dedicated A2DP data link is required between the baseband processor and Bluetooth transceiver for streaming music to a stereo Bluetooth headset. The LM49370 can be bypassed during A2DP operation.

However, the LM49370 has built-in amplifiers to drive a wired pair of stereo headphones for non-Bluetooth applications.

## Revision History

| Rev | Date | Description |
|---|---|---|
| 1.0 | 07/17/07 | Initial release. |



**National Semiconductor Americas Customer Support Center**
Email: new.feedback@nsc.com
Tel: 1-800-272-9959

**National Semiconductor Europe Customer Support Center**
Fax: +49 (0) 180-530-85-86
Email: europe.support@nsc.com
Deutsch Tel: +49 (0) 69 9508 6208
English Tel: +49 (0) 870 24 0 2171
Français Tel: +33 (0) 1 41 91 8790

**National Semiconductor Asia Pacific Customer Support Center**
Email: ap.support@nsc.com

**National Semiconductor Japan Customer Support Center**
Fax: 81-3-5639-7507
Email: jpn.feedback@nsc.com
Tel: 81-3-5639-7560