CML Microcircuits

COMMUNICATION SEMICONDUCTORS

Application Note

CMX649 Operation and Application

AN/2WR/649/2 February 2013

Additional Resources	None

Introduction

The CMX649 is an extremely flexible voice codec with an extensive range of available features. The purpose of this application note is to help the reader understand how to use the CMX649 by providing information beyond that which is available in the CMX649 data sheet.

The CMX649 data sheet should be consulted when reviewing this application note. This application note also presumes that the reader has a basic understanding of voice coding.

History

Version	Changes	Date
2.0	XTAL/CLOCK low frequency input limits included	20-2-13
1.0	Original Release	27-6-03

TABLE OF CONTENTS

1 Timing Considerations 3
1.1 XTAL/CLK Input 3
1.2 Switched Capacitor Filter Clock Considerations 4
1.3 Encode/Decode Bit Clock Rate Determination 6
1.3.1 Encode Bit Clock 6
1.3.1.1 Encode Bit Clock from TX CLK Pin 7
1.3.1.2 Encode Bit Clock Internally Derived from XTAL/CLK Pin 8
1.3.1.3 Encode Bit Clock Internally Supplied From Decode Bit Clock 9
1.4 Decode Bit Clock 9
1.4.1 Decode Bit Clock from RX CLK Pin 10
1.4.1.1 Decode Bit Clock Internally Derived from XTAL/CLK Input 11
1.4.1.2 Decode Bit Clock Internally Derived from Incoming Rx Data 11
1.4.1.3 Decode Bit Clock Internally Adjusted by STROBE Input 12
2 Microprocessor Interface Considerations 13
2.1 C-BUS Interface 13
2.2 Burst Mode 13
2.2.1 Encode Burst Mode Operation: 14
2.2.2 Decode Burst Mode Operation 15
3 Power Control 17
4 Voice Activity Detector 19
4.1 Attack and Decay Time Constant Programming 20
4.2 Threshold Level Programming 20
4.3 VAD Output Level 20
5 Audio Path Considerations 22
5.1 Microphone Amplifier 22
5.2 Audio Input Digitally Controlled Amplifier (DCA) 23
5.3 Programmable Anti-Alias Filter (AAF) 23
5.4 Sidetone DCA 24
5.5 Programmable Anti-Image Filter (AIF) 24
5.6 Volume DCA 24
6 Voice Encoding and Decoding 25
6.1 General Comments 25
6.2 Encoder Specific Considerations 25
6.3 Decoder Specific Considerations 26
6.4 Voice Coding Scheme 28
6.4.1 ADM 28
6.4.1.1 Companding Rule Selection 28
6.4.1.2 Maximum and Minimum Step Height Selection 29
6.4.1.3 Integrator Time Constant Selection 30
6.4.1.4 Second Order Integration 30
6.4.1.5 CVSD 31
6.4.2 Linear PCM 31
6.4.2.1 Suggestions For Linear PCM Operation 31
6.4.3 Non-Linear PCM 33
6.4.3.1 A-Law PCM 33
6.4.3.2 μ-Law PCM 33
7 Transcoding 34
7.1 Transcoding with ADM Input 35
7.2 Transcoding with PCM Input 36
8 Conclusion 39

1 Timing Considerations

A discussion of the capabilities of the CMX649 should logically start with an overview of the timing requirements of the chip. After all, the device will not operate properly unless its timing requirements are met.

The CMX649 has an internal clock generator section that develops all necessary timing signals for the device. This clock generator can be simplified into the following diagram (please note that “SCF Clock” refers to the internally derived switched-capacitor filter clock):

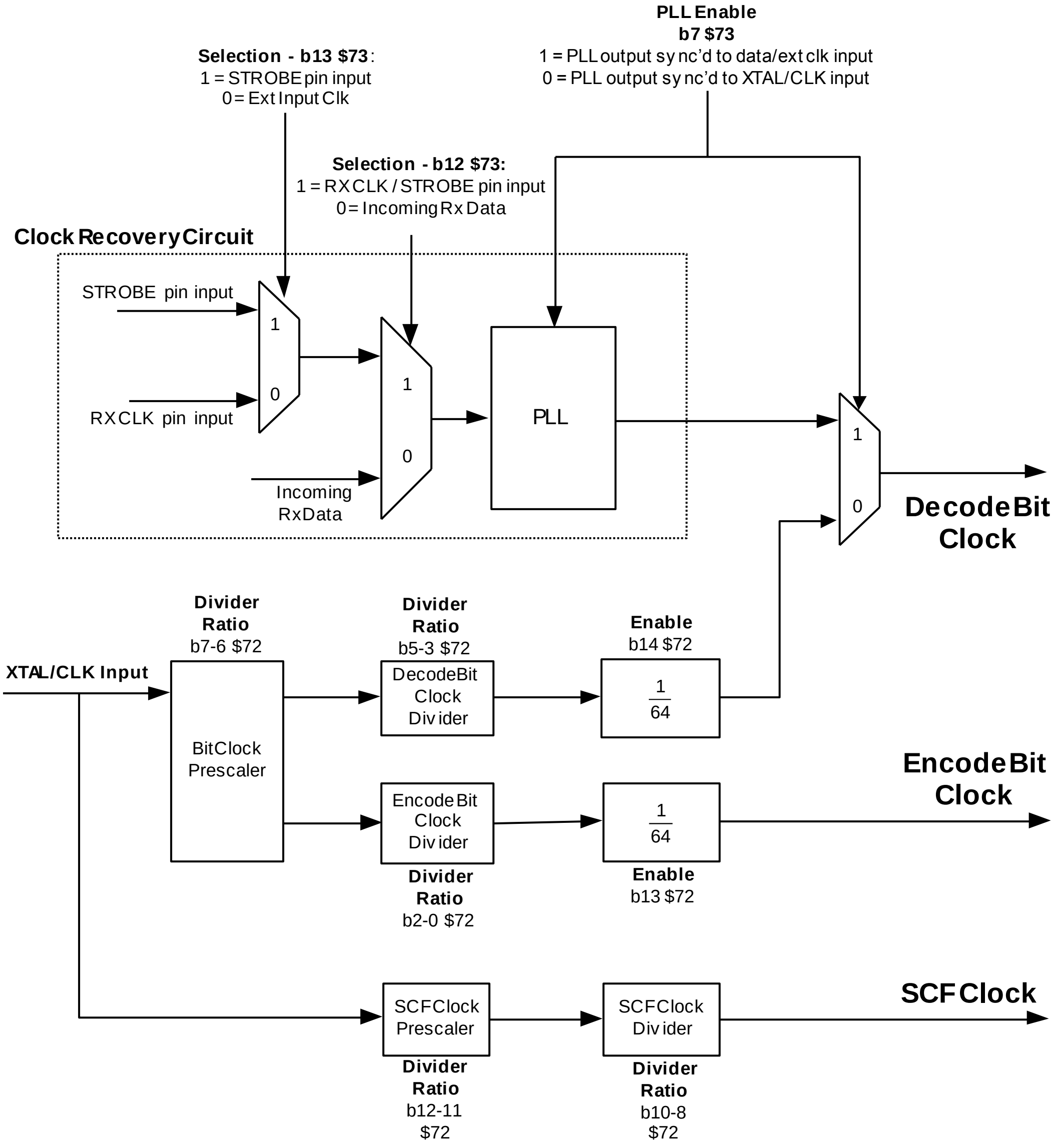

Figure 1: Internal Clock Generator

Details about the various portions of the internal clock generator are provided throughout this section of the application note.

1.1 XTAL/CLK Input

The first timing source to consider is the “master timing source” for the CMX649, namely, its Xtal/Clock oscillator input. The CMX649 can accept a range of frequencies from 2.048MHz to 16.384MHz, so a wide range of frequencies are supported.

It should be noted that due to the low power design of the oscillator circuit Xtals should only be used that operate over 8MHz. For operation below 8MHz a buffered clock should be used.

The CMX649 can also use a clock signal already in use in the circuit, so long as it is one of the acceptable frequencies listed in the CMX649 data sheet. When this approach is used, the clock signal should be DC coupled into the XTAL/CLK pin of the CMX649.

Please note that a timing source MUST be applied to the XTAL/CLK pin, regardless of how the encode/decode bit clocks (i.e. sampling clocks) are derived.

Regardless of where the timing signal comes from, the XTAL POWER SAVE bit (b1) of the POWER CONTROL 2 register ($65) must be set to a 0 in order for the CMX649 oscillator circuit to function properly.

1.2 Switched Capacitor Filter Clock Considerations

The switched capacitor filter (SCF) architecture of the CMX649 was designed to operate with an internally derived 256kHz clock frequency. This SCF clock is derived from the input to the XTAL/CLK pin.

While the XTAL/CLK input can be one of many different frequencies, the CMX649 internal SCF clock frequency must be maintained as close to 256kHz as possible; otherwise, the programmed anti-alias filter (AAF) and anti-image filter (AIF) bandwidth values will scale proportionately.

The CLK DIVIDER CONTROL register ($72) can be adjusted to achieve the 256kHz SCF clock frequency for various XTAL/CLK input signals. In general:

$$\text{SCF Clock Frequency} = \frac{\text{XTAL/CLK input}}{\text{prescaler x divider}}$$

where:

- prescaler = “divider ratio” obtained from b12-11 of CLK DIVIDER CONTROL register ($72)
- divider = “divider ratio” obtained from b10-8 of CLK DIVIDER CONTROL register ($72)

The filter clock prescaler must be enabled in order to derive the SCF clock from the XTAL/CLK input, and this is done with bit 15 of the CLK DIVIDER CONTROL register ($72).

The following table illustrates how the CLK DIVIDER CONTROL register ($72) can be adjusted to achieve the 256kHz SCF clock frequency for all of the available XTAL/CLK input frequencies.

SCF Clock Frequency (in kHz)									
XTAL/CLK Freq (MHz)	Resulting Filter Clock Prescaler “divider ratio”	Resulting Filter Clock “divider ratio”	8	15.5	15.75	16	22	31.25	47.75
		Filter Clock Divider bits[10:8] Filter Clock Prescaler bits [12:11]	001	010	011	100	101	110	111
2.048	1	00	256.000						
4.000	1	00		258.065	253.968				
4.032	1	00			256.000				

4.096	1	00				256.000			
8.000	1	00						256.000	
8.064	2	01			256.000				
8.192	2	01				256.000			
11.2896	2	01					256.582		
12.000	1	00							256.684
12.096	3	10			256.000				
12.288	3	10				256.000			
16.000	2	01						256.000	
16.128	4	11			256.000				
16.384	4	11				256.000			

Table 1: Relationships that Determine SCF Clock Frequency

As can be seen from the above chart, not all of the allowable XTAL/CLK inputs will achieve exactly the necessary 256kHz SCF clock. SCF clock values that aren’t equal to 256kHz will cause a proportionate scaling in the programmed AAF and AIF frequencies.

For example, a 12MHz XTAL/CLK input will only achieve a 256.684kHz SCF clock. This deviation from the target SCF clock frequency will cause the programmed AAF and AIF frequency values to increase by 0.3%. Acceptable device operation with this SCF clock is certainly possible, but the user needs to be aware of this when setting the AAF and AIF bandwidth values.

Some power savings can be realised by using a higher prescaler value in conjunction with a lower divider value, as this will lower the prescaler output frequency.

The filter passband and stopband corner frequencies specified in the CMX649 specifications are based on the use of 4.096MHz, 8.192MHz, 12.288MHz, or 16.384MHz XTAL/CLK input, with appropriately adjusted prescaler and divider values. **For other XTAL/CLK inputs, the CMX649 filter passband and stopband corner frequencies will scale proportionately.**

NOTE: while the CMX649 allows a wide range of XTAL/CLK inputs, **not all sampling rates can be achieved with any of the allowed XTAL/CLK inputs**. The following table lists the minimum required XTAL/CLK input frequencies for various data sampling rates.

Desired Data Sampling Rate	**Minimum XTAL/CLK input for desired data rate**
32kbps	2.096MHz
64kbps	4.096MHz
128kbps	8.192MHz

Table 2: Sampling Rate versus Required XTAL/CLK Input

1.3 Encode/Decode Bit Clock Rate Determination

The CMX649 uses a “bit clock” (i.e. sampling clock) to encode and decode voice. Encode and decode bit clocks can be internally generated by the CMX649 clock generator section, or they can be supplied from external sources. The optimal source for encode and decode bit clocks is application dependent.

1.3.1 Encode Bit Clock

The encode bit clock (i.e. sampling clock) can be supplied by three different sources in accordance with the following table:

CLK SOURCE CONTROL ($73)		**Encode Bit Clock Source**
Bit 5	Bit 4	
x	0	Externally applied to TX CLK pin
0	1	Internally derived from XTAL/CLK input (this selection is required for burst mode operation)
1	1	Internally supplied from decode bit clock

Table 3: Encode Bit Clock Sources

IMPORTANT NOTES:

- XTAL/CLK input is always required, regardless of the source of the decode bit clock.
- the same bit clock frequency must be used for both encode and decode operation.

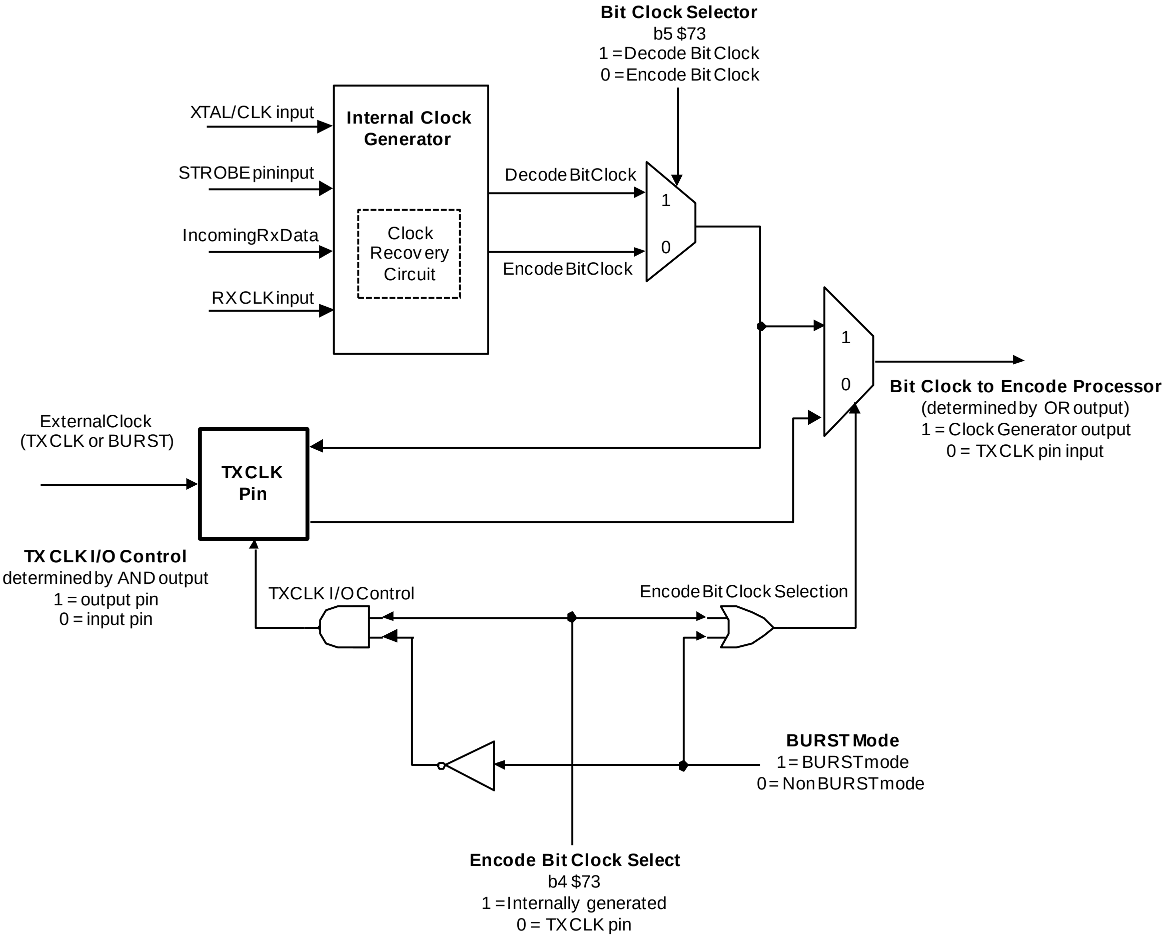

Figure 2: Encode Bit Clock Sources

(Please refer to Section 2.2 for a description of “burst” mode operation.)

1.3.1.1 Encode Bit Clock from TX CLK Pin

Some applications may benefit from having the encode bit clock provided by a source external to the CMX649. In this type of application:

- The encode bit clock signal should be DC coupled to the TX CLK pin.
- The TX CLK pin must be configured as an input pin (i.e. $73 bit 4 = 0)

When the encode bit clock is supplied on the TX CLK pin, the TX CLK pin input frequency must not exceed 1/60th of the output of the internal bit clock prescaler. (The output of the internal bit clock prescaler is a function of the XTAL/CLK input signal and the prescaler division ratio.)

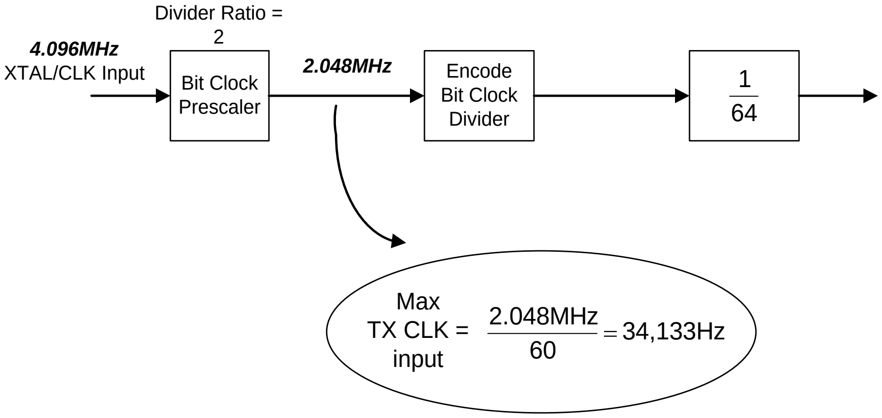

Figure 3: Example of Maximum TX CLK Input Determination

The following table provides some examples of XTAL/CLK inputs and the corresponding maximum TX CLK input frequencies. This table does not represent all possible XTAL/CLK inputs or prescaler division ratios, but is intended to provide guidance for users wishing to calculate the maximum TX CLK frequencies in their applications:

XTAL/CLK Input Frequency	**Internal Bit Clock Prescaler Division Ratio** (b7-6, CLK DIVIDER CONTROL register $72)	**Maximum Bit Clock Frequency on TX CLK Pin (approximate)**
2.096MHz	1	34.9kHz
4.032MHz	1	67.2kHz
8.192MHz	2	68.2kHz
12.288MHz	3	68.2kHz
16.384MHz	4	68.2kHz

Table 4: Maximum TX CLK Frequency versus XTAL/CLK Input

1.3.1.2 Encode Bit Clock Internally Derived from XTAL/CLK Pin

In this operating condition, the encode bit clock is internally derived from the XTAL/CLK input signal.

NOTE: the encode bit clock must be derived from the XTAL/CLK input signal for burst mode operation.

The encode bit clock is enabled with b13 in the CLK DIVIDER CONTROL register ($72) to operate in this mode. This register can also be adjusted to provide a wide variety of encode bit clock rates from the XTAL/CLK input signal. In general:

$$\text{Encode Bit Clock Frequency} = \frac{\text{XTAL/CLK input}}{\text{prescaler x divider x 64}}$$

where:

- prescaler = “divider ratio” obtained from b7-6 of CLK DIVIDER CONTROL register ($72)
- divider = “divider ratio” obtained from b2-0 of CLK DIVIDER CONTROL register ($72)

1.3.1.3 Encode Bit Clock Internally Supplied From Decode Bit Clock

Encode (and decode) bit clocks must be provided at transmit and receive locations. Most codec solutions require a precise timing source for the bit clock signal at both the encode (e.g. cordless telephone base unit) and decode (e.g. cordless telephone handset unit) locations. This approach adds cost to the end product.

The CMX649 provides an on-chip “data clock recovery” circuit that can recover the timing signal used in the original encode process (i.e. extracts the clock from the incoming data stream). This feature minimises cost and improves audio quality by:

- Allowing the use of a less expensive timing source at the handset for the XTAL/CLK input.
- Providing a precise encode/decode bit clock from the recovered data stream.

The CLK SOURCE CONTROL register ($73) must be properly configured to support this operating mode. Please refer to Section 1.4.1.2, “Decode Bit Clock Synchronised to Rx Data”, for guidance on how to properly configure the CMX649 for this operating mode.

NOTE: The encode bit clock will be identical to the decode bit clock in this operating mode. The flexibility of the CMX649 allows the decode bit clock to be derived from a source OTHER than the clock recovery circuit. If the user wishes for the encode bit clock to be derived from the incoming Rx data signal, the user must properly configure the decode bit clock source for this to occur.

1.4 Decode Bit Clock

The decode bit clock can be supplied from four different sources in accordance with the following table:

CLK SOURCE CONTROL register ($73)				**Decode Bit Clock Source**
B13	**B12**	**B7**	**B6**	
X	X	X	0	Externally applied to RX CLK pin
X	X	0	1	Internally derived from XTAL/CLK input (this selection is required for burst mode operation)
X	0	1	1	Internally derived from incoming Rx data
1	1	1	1	Internally adjusted by STROBE input

Table 5: Decode Bit Clock Sources

IMPORTANT NOTES:

- XTAL/CLK input is always required, regardless of the source of the decode bit clock.
- the same bit clock frequency must be used for both encode and decode operation.

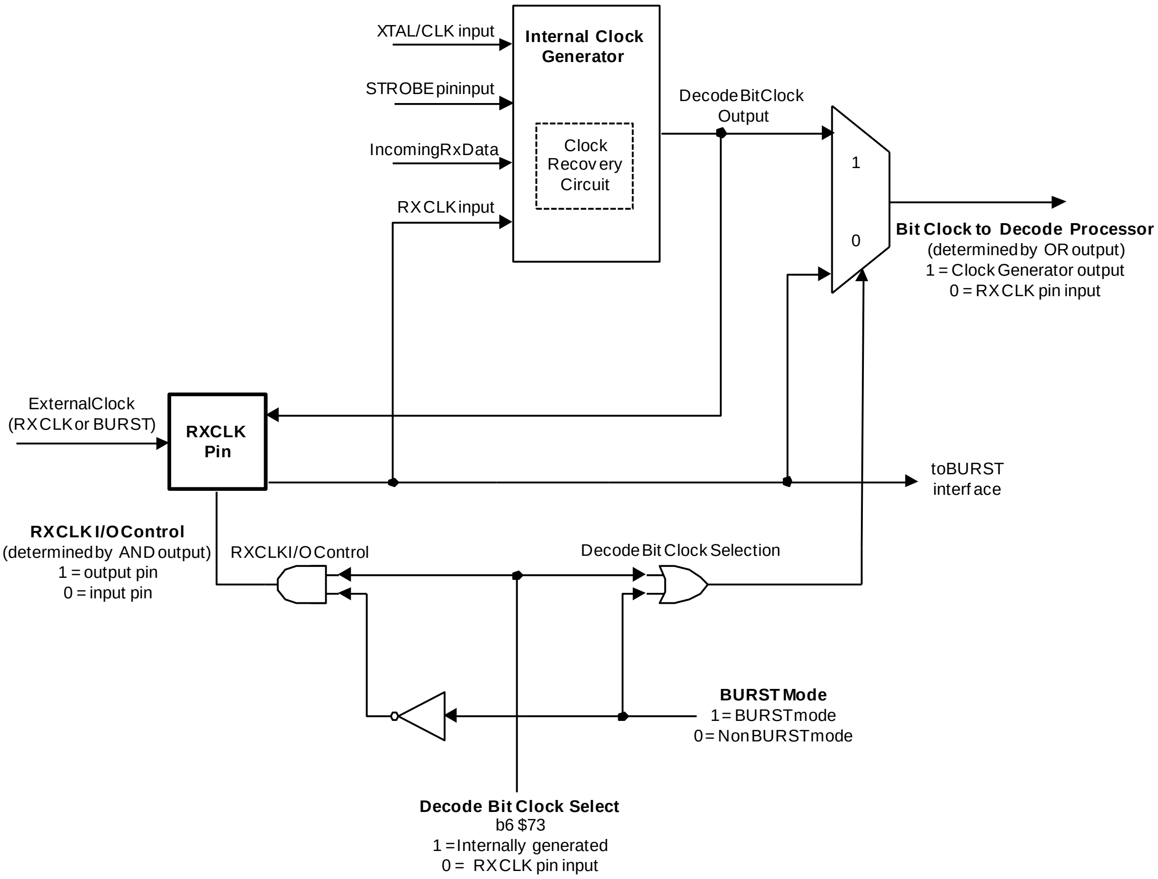

Figure 4: Decode Bit Clock Sources

(Please refer to Section 2.2 for a description of “burst” mode.)

1.4.1 Decode Bit Clock from RX CLK Pin

Some applications may benefit from having the decode bit clock provided directly by a source external to the CMX649. In this type of application, the decode bit clock signal should be DC coupled to the RX CLK pin.

Since both the RX CLK and Rx data signals are being provided from a source external to the CMX649, these signals should be appropriately synchronised when this operating mode is used.

The RX CLK pin input frequency must not exceed 1/60th of the output of the internal bit clock prescaler. (The output of the internal bit clock prescaler is a function of the XTAL/CLK input signal and the prescaler division ratio.)

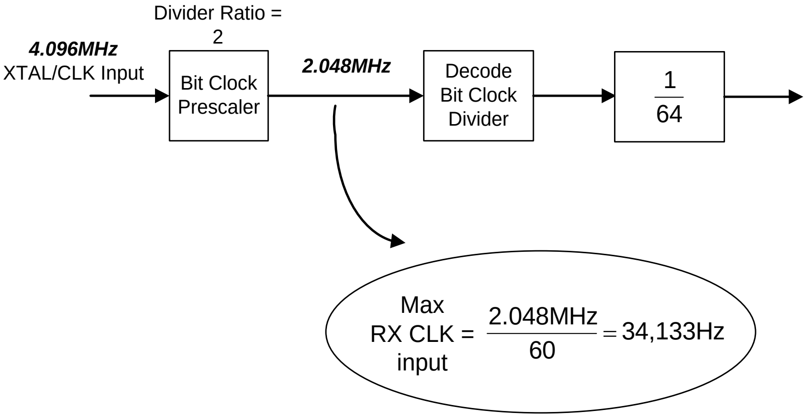

Figure 5: Example of Maximum RX CLK Input Determination

The following table provides some examples of XTAL/CLK inputs and the corresponding maximum RX CLK input frequencies. This table does not represent all possible XTAL/CLK inputs or prescaler division ratios, but is intended to provide guidance for users wishing to calculate the maximum RX CLK frequencies in their applications:

XTAL/CLK Input Frequency	Internal Bit Clock Prescaler Division Ratio (b7-6 CLK DIVIDER CONTROL register $72)	Maximum Bit Clock Frequency on RX CLK Pin (approximate)
2.096MHz	1	34.9kHz
4.032MHz	1	67.2kHz
8.192MHz	2	68.2kHz
12.288MHz	3	68.2kHz
16.384MHz	4	68.2kHz

Table 6: Maximum RX CLK Frequency versus XTAL/CLK Input

1.4.1.1 Decode Bit Clock Internally Derived from XTAL/CLK Input

In this operating condition, the decode bit clock is internally derived from the XTAL/CLK input signal.

The decode bit clock must be derived from the XTAL/CLK input signal for burst mode operation.

The decode bit clock is enabled with bit 14 in the CLK DIVIDER CONTROL register ($72). This register can also be adjusted to provide a wide variety of decode bit clock rates from the available XTAL/CLK inputs. In general:

$$\text{Decode Bit Clock Frequency} = \frac{\text{XTAL/CLK input}}{\text{prescaler x divider x 64}}$$

where:

- prescaler = “divider ratio” obtained from b7-6 of CLK DIVIDER CONTROL register ($72)
- divider = “divider ratio” obtained from b5-3 of CLK DIVIDER CONTROL register ($72)

1.4.1.2 Decode Bit Clock Internally Derived from Incoming Rx Data

To achieve optimal recovered voice quality, the bit clock used to sample the Rx data should be identical to that used in the original encoding process. In other words, the original Tx data clock should be used to sample the Rx data stream.

Other voice codec solutions require the use of an external circuit to extract the Tx data clock from the Rx data stream. This approach increases component cost, board space, and power consumption for the end application.

The CMX649 includes an on-board data clock recovery circuit, which means great audio quality can be achieved at a fraction of the cost of competing solutions.

The clock recovery circuit PLL can be locked to several different signals, so the CLK SOURCE CONTROL register ($73) must be properly configured, in accordance with Table 5, to support this operating mode.

1.4.1.3 Decode Bit Clock Internally Adjusted by STROBE Input

The externally applied STROBE clock signal can be used in two ways:

- Non-burst mode; pulses on the STROBE input will adjust the timing of the internally derived decode bit clock, which will help maintain optimal phase relationship with external timing requirements.
- Burst mode: “sync” pulses are provided on the STROBE input to mark byte boundaries.

2 Microprocessor Interface Considerations

2.1 C-BUS Interface

The CMX649 uses a simple serial microcontroller (µC) interface known as “C-BUS”. This µC interface, which was developed by CML, is very similar to other serial interfaces such as SPI.

The CMX649 C-BUS interface consists of five lines:

- SCLK: Serial Clock, provided by the host µC, that synchronises all C-BUS transactions
- CMD: Command Data, the control information passed from the host µC to the CMX649
- RPLY: Reply Data, the data/control information passed from the CMX649 to the host µC
- CSN: Chip Select (active low), used to indicate which C-BUS part is currently active
- IRQN: Interrupt Request (active low), used to indicate when the CMX649 requires host µC servicing.

The communication of information between the CMX649 and the host µC over the C-BUS interface is typically referred to as a C-BUS “transaction”. In general, a single C-BUS transaction flows as follows:

- CSN is taken low.
- An “Address/Command” byte (A/C byte) is passed to the CMX649 over the CMD line. This information indicates which CMX649 register is to be addressed.
- If a CMX649 “write” register was specified in the A/C byte, one or two configuration bytes (depends on register size) will be sent from the host µC to the CMX649 over the CMD line.
- If the A/C byte included a “read” register address, the contents of that register (one or two bytes, depending on register size) will then be passed back to the host µC over the RPLY line.
- The CSN line is then taken high again to complete the transaction.

More information on the C-BUS interface can be found at: ***http://www.cmlmicro.com/products/applications/C-BUS_2.pdf***

2.2 Burst Mode

The CMX649 provides two means of data exchange for Tx data output & Rx data input; burst mode and non-burst mode. The burst mode allows data to be exchanged with the CMX649 in eight-bit or sixteen-bit bytes, while the non-burst mode causes the CMX649 to process data (encode and decode) one bit at a time.

Bits 2-0 of the CODEC MODE CONTROL register ($70) determine whether the CMX649 operates in eight-bit burst mode, sixteen-bit burst mode, or non-burst mode. All burst modes are eight-bit with the exception of “Linear PCM with buffered I/O” mode, which is a sixteen-bit scheme. (Note: The lowest three bits of the sixteen-bit PCM output word should be ignored in “Linear PCM with buffered I/O” mode.)

The following table indicates the data exchange mode used by each of the operating modes of the CMX649:

CODEC MODE CONTROL Register ($70)			**CODEC Mode**	**Data Exchange Mode**
Bit 2	**Bit 1**	**Bit 0**		
0	0	0	ADM mode without buffered I/O	Non-burst
0	0	1	ADM mode with buffered I/O	8-bit burst
0	1	0	Linear PCM with buffered I/O	16-bit burst
0	1	1	µ-law PCM with buffered I/O	8-bit burst
1	0	0	A-law PCM with buffered I/O	8-bit burst

Table 7: Codec Operating Modes

In general, burst mode operation relies primarily on three timing signals:

- SYNC pulse: applied to the STROBE pin, used to mark byte boundaries
- BURST CLOCK: applied to the RX CLK pin, used to clock in/out individual data bits.
- Bit Clock: derived internally for both Tx and Rx operation.

The “sync” pulse should be applied on the STROBE pin every eight or sixteen bit times, depending on the selected operating mode.

The TX CLK and RX CLK pins are automatically forced to become inputs during burst mode operation. **The BURST CLOCK signal must be applied to the RX CLK pin during burst mode operation, regardless of whether encode or decode operation is performed.**

Burst mode operation requires that the encode and decode bit clocks be internally derived from the XTAL/CLK input. Please refer to Section 1.3, “Encode/Decode Bit Clock Rate Determination”, for information on proper configuration of encode/decode bit clocks in this mode.

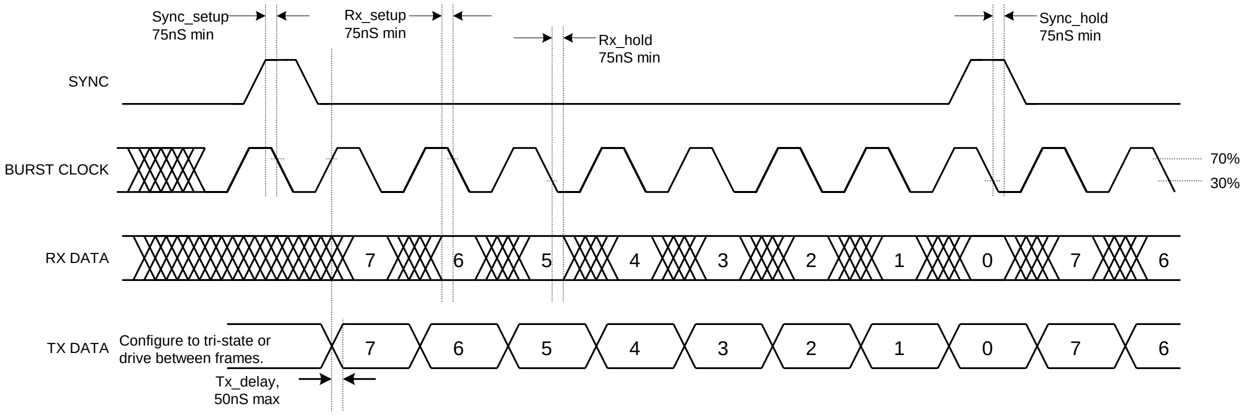

Figure 6: Burst Mode Timing Diagram

2.2.1 Encode Burst Mode Operation:

The BURST CLOCK signal, which must be externally applied to the RX CLK pin, clocks the encode data out of the TX DATA pin during burst mode operation.

The encode bit clock should be internally derived from the XTAL/CLK input during burst mode operation. Please refer to Section 1.3.1, “Encode Bit Clock”, for information on proper configuration of encode/decode bit clocks in this mode.

ADM or PCM output data is transferred from the encode processor to the TX DATA pin for processing by the host microcontroller. The encoded data transitions on the BURST CLOCK rising edge and is valid on the BURST CLOCK falling edge.

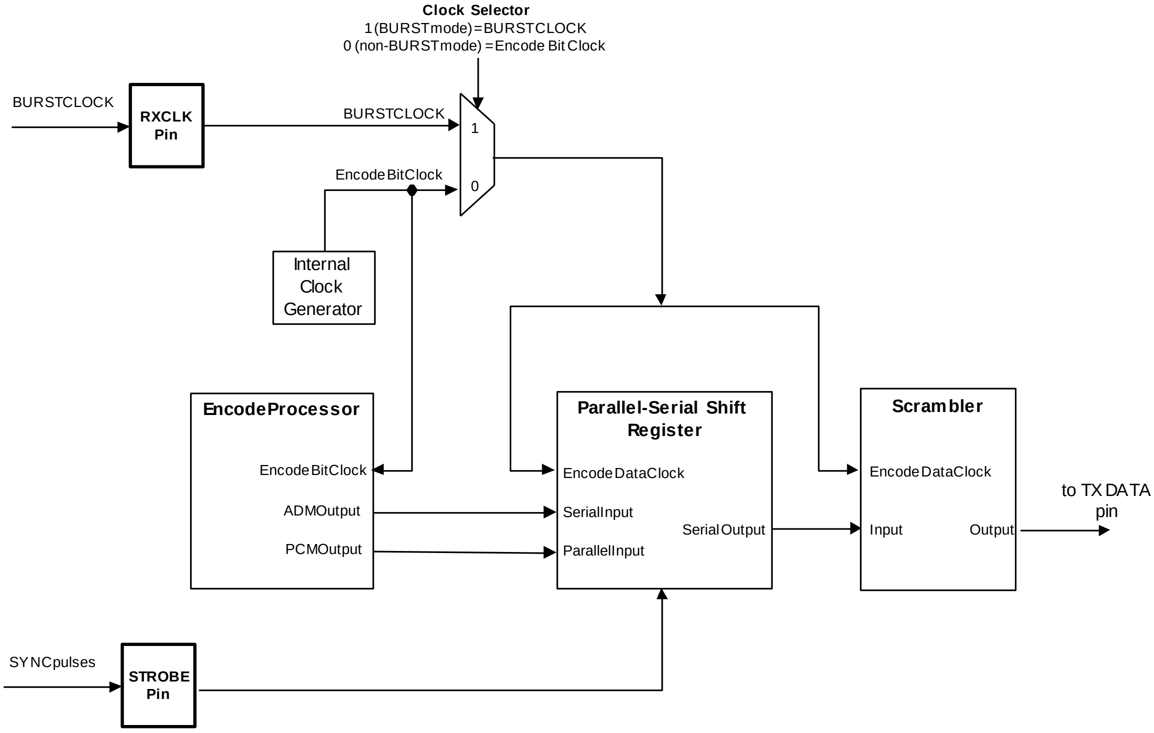

Figure 7: Encode Burst Mode Operation

2.2.2 Decode Burst Mode Operation

The BURST CLOCK signal, which must be externally applied to the RX CLK pin, clocks the data into the RX DATA pin during burst mode operation.

The decode bit clock should be internally derived from the XTAL/CLK input during burst mode operation. Please refer to Section 1.4, “Decode Bit Clock”, for information on proper configuration of encode/decode bit clocks in this mode.

Input data is clocked in to the descrambler by the BURST CLOCK signal. The descrambler output is fed to the serial-parallel shift register (S/P SR). The first BURST CLOCK falling edge after the SYNC rising edge will cause the previously loaded eight or sixteen bits to be latched into the decode processor. The next BURST CLOCK falling edge will cause the first bit of the next byte to be clocked into the descrambler and S/P SR.

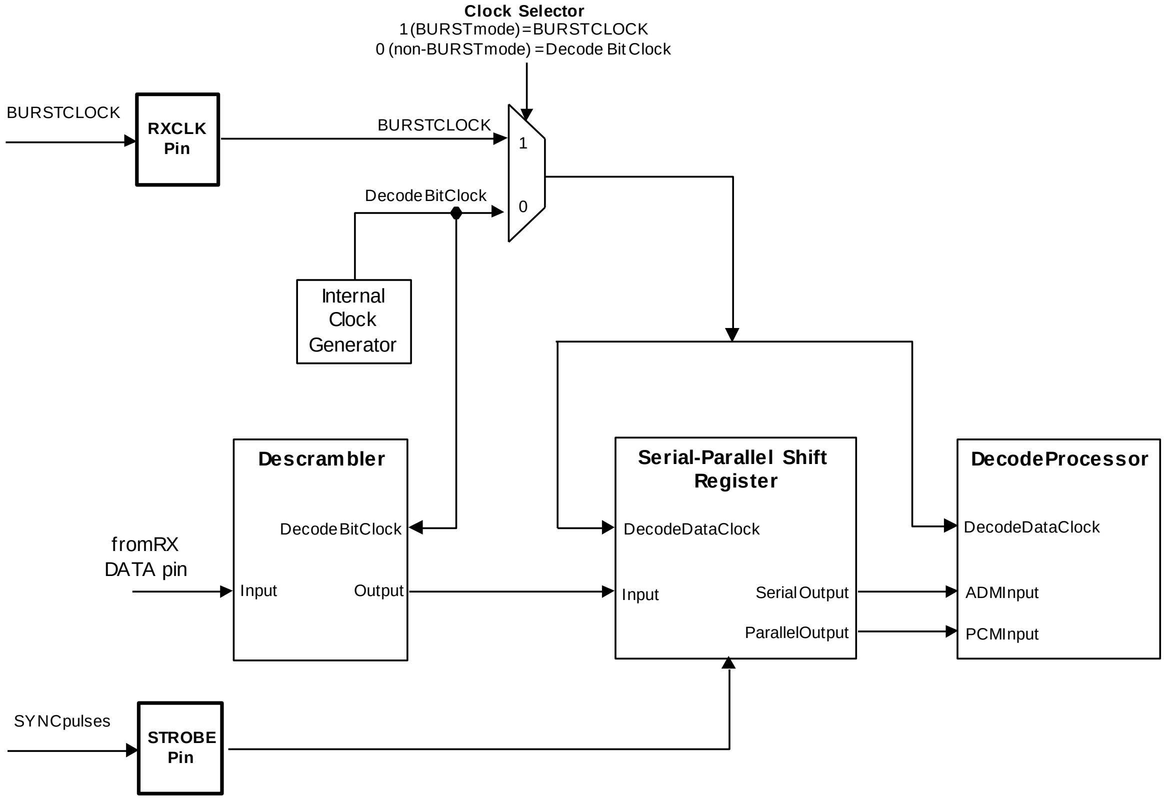

Figure 8: Decode Burst Mode Operation

3 Power Control

The CMX649 offers the ability to power down many different sections of the chip. Additionally, the CMX649 also provides reduced power consumption operating modes for several sections of the device. These features allow the design engineer to develop optimal power consumption strategies for specific applications.

The CMX649's power control capabilities are harnessed primarily through two eight-bit registers: POWER CONTROL 1 ($64) and POWER CONTROL 2 ($65). These registers adjust the power consumption settings for the following functions:

- Anti-Aliasing Filter
- Anti-Imaging Filter and Sidetone Digitally Controlled Amplifier (DCA)
- Encode DAC, which includes:
  - Smoothing filter prior to encode comparator
  - Encode comparator
- Decode DAC
- Microphone Amplifier
- Audio Input DCA
- Volume DCA
- Crystal Oscillator Circuit
- Entire Analogue Section

In general, significant power savings can be achieved simply by deactivating circuits that aren't being used. For example, here are some functions that can be powersaved for encode and decode operation. Please note that the actual functions available for power savings are application dependent:

Encode Operation		
Function to be Powersaved	Controlling Register	Register Bits
Anti-Imaging Filter	$64	5-4
Decode DAC (analogue section of decoder)	$64	1-0
Volume DCA (digitally controlled amplifier)	$65	3-2
De-scrambler	$71	12
Decode bit clock	$72	14
PLL (clock recovery circuit)	$73	7
Data filter and slicer	$73	2-1
Decoder (digital section)	$81	3

Table 8: Possible Powersave Sections for Encode Operation

Decode Operation		
Function to be Powersaved	Controlling Register	Register Bits
Anti-Alias Filter	$64	7-6
Encode DAC (analogue section of encoder)	$64	3-2
Mic amp	$65	7-6
Audio DCA (digitally controlled amplifier)	$65	5-4
Scrambler	$71	15
Encode bit clock	$72	13
Encoder (digital section)	$81	7

Table 9: Possible Powersave Sections for Decode Operation

Certain CMX649 sections provide power control settings that are based on the data rate and codec bandwidth desired for the application. In general, lower data rates and smaller codec bandwidths provide for incrementally larger power savings.

Proper configuration of CMX649 timing parameters can also yield power savings. Using a higher prescaler value in conjunction with a lower divider value will lower the prescaler output frequency, which will result in power savings. Please refer to Section 1, “Timing Considerations”, for more explanation of these options.

4 Voice Activity Detector

The CMX649 provides a voice activity detector (VAD) for both encode and decode operation. This function can be used to detect periods of voice inactivity so that unneeded circuits can be powersaved, thus improving battery life and customer satisfaction.

Certain coding schemes, such as CVSD, can suffer from added noise (e.g. granular noise) when low level input signals are encoded. The VAD function can minimise this by alerting the microcontroller to low level input signals. The microcontroller can then take action to quiet the speaker output by:

- Disabling the speaker.
- Load an “idling” pattern into the CMX649 which will result in a quiet speaker output.

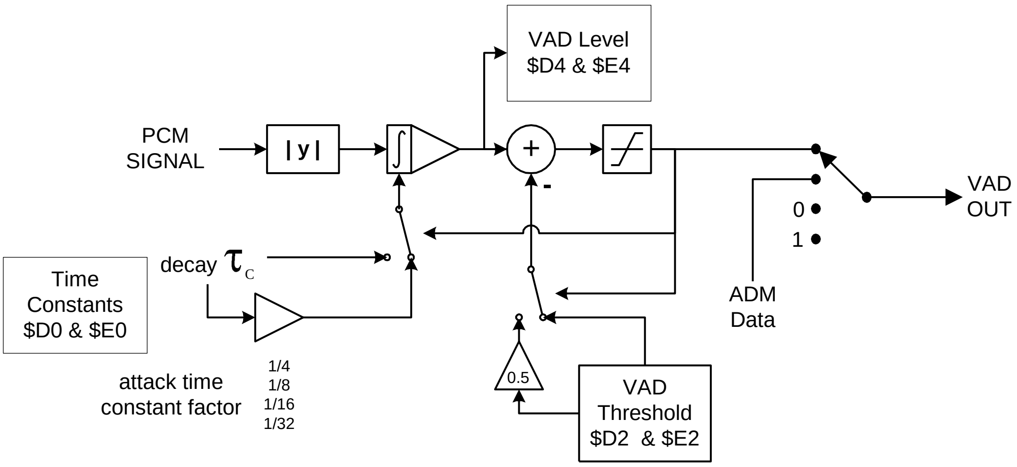

Table 10: VAD Block Diagram

The VAD function is implemented with an energy detector circuit consisting of an absolute value function, an integrator and a threshold detector. The input to the VAD (PCM signal) is rectified and averaged with a lossy integrator. The integrator output is compared to the VAD threshold to derive the “VAD_OUT” logic signal, which is then presented on the ENC VAD and DEC VAD pins:

VAD_OUT = 1 $\Rightarrow$ signal energy greater than the threshold is present.
VAD_OUT = 0 $\Rightarrow$ signal energy is below the threshold.

The threshold detector levels are programmable via the DECODE ($D2) and ENCODE ($E2) VAD THRESHOLD registers. The integrator time constants (i.e. attack and decay time) are independently controlled through the DECODER ($D0) and ENCODER ($E0) MODE AND SETUP registers. Typical attack and decay times used for detecting voice activity are 5ms and 150ms, respectively.

The energy levels detected by the VAD may be read from the DECODE and ENCODE VAD LEVEL OUTPUT Registers ($D4 and $E4). This information can be used to adaptively set the detector threshold value by observing the energy level of background noise.

A summary of the CMX649 internal registers and relevant bits for VAD operation are provided in the table below:

VAD Function	Encode		Decode	
	Register	Bits	Register	Bits
Attack Time Constant	$E0	4-3	$D0	4-3
Decay Time Constant	$E0	7-5	$D0	7-5
VAD OUT Output Source	$E0	2-1	$D0	2-1
Threshold Detection Level	$E2	15-0	$D2	15-0
Integrator Output Level	$E4	15-0	$D4	15-0

Table 11: VAD Control Registers

4.1 Attack and Decay Time Constant Programming

The attack and decay time constants are configured in the ENCODER ($E0) and DECODER ($D0) MODE AND SETUP registers. These time constants determine the time interval over which the rectified PCM input signal is integrated.

The attack time constant is derived from the decay time constant in accordance with bits 4-3 of $E0 (encode) and $D0 (decode). Consequently, the decay time constant must first be selected in order to properly configure the attack time constant.

A longer attack/decay time constant will improve reliability by reducing the probability that noise will cause a false positive VAD OUT signal. A tradeoff exists, however, between time constant length and accuracy of voice detection. If the input signal is below the detection threshold, a longer attack/decay time constant will create a longer time period before the VAD OUT signal indicates the inactivity. This will force the circuitry to remain powered for longer than is actually needed. Therefore, the improved robustness of a longer time constant comes at the expense of shorter battery life.

Typical values for attack and decay time constants are 5ms and 150ms, respectively, but optimal values for these time constants are application dependent.

4.2 Threshold Level Programming

The desired VAD threshold voltage level is programmed into the DECODE ($D2) and ENCODE ($E2) VAD THRESHOLD registers with the following equation:

$$\text{Register Value} = \frac{(\text{Desired Threshold Voltage}) \cdot 32768}{(\text{DAC Full Scale Reference Voltage})}$$

where:

- Register Value = binary representation of quotient
- Register value can vary between $0 to $7FFF (0 to 32767).
- DAC Full Scale Reference Voltage = 1.25V

For example, to compute the required register value for a VAD threshold of 100mV:

$$\text{Register Value} = \frac{100\text{mV} \cdot 32768}{1.25} = 2621.44 = \$0\text{A3D}$$

4.3 VAD Output Level

The VAD integrator output voltage level, which corresponds to the time-averaged PCM input signal, can be monitored through the DECODE ($D4) and ENCODE ($E4) VAD LEVEL OUTPUT READ registers. The following equation can be used to interpret the results contained in these sixteen-bit read-only registers:

$$\text{VAD Output Level} = \frac{(\text{Register Value}) \cdot \text{DAC Full Scale Reference Voltage}}{-32768}$$

where:

- Register Value = decimal representation of contents of $D4 or $E4
  - register contents are in “negative 2’s complement” format
  - register contents can range from $0 to $8000 (0 to –32768)
- DAC Full Scale Reference Voltage = 1.25V

For example, to compute the VAD output level corresponding to $3AB0 in register $E4:

$$\text{VAD Output Level} = \frac{(\text{-15024}) \cdot 1.25}{-32768} = 0.573V$$

The VAD output level information can be used to dynamically adjust the VAD threshold to account for varying background noise, if desired.

The VAD OUT output source can be selected from a list of four options. These options are defined by b2-1 of $E0 (encode) and $D0 (decode) registers and include:

- VAD Threshold Detector Output (normal operation).
- ADM data bits at the ADM data rate.
- Forced to logic high condition.
- Forced to logic low condition.

5 Audio Path Considerations

The CMX649 provides considerable filtering and amplification options on-chip, thereby reducing the number of external components and bill-of-materials cost.

The audio path functions provided by the CMX649 are illustrated overleaf:

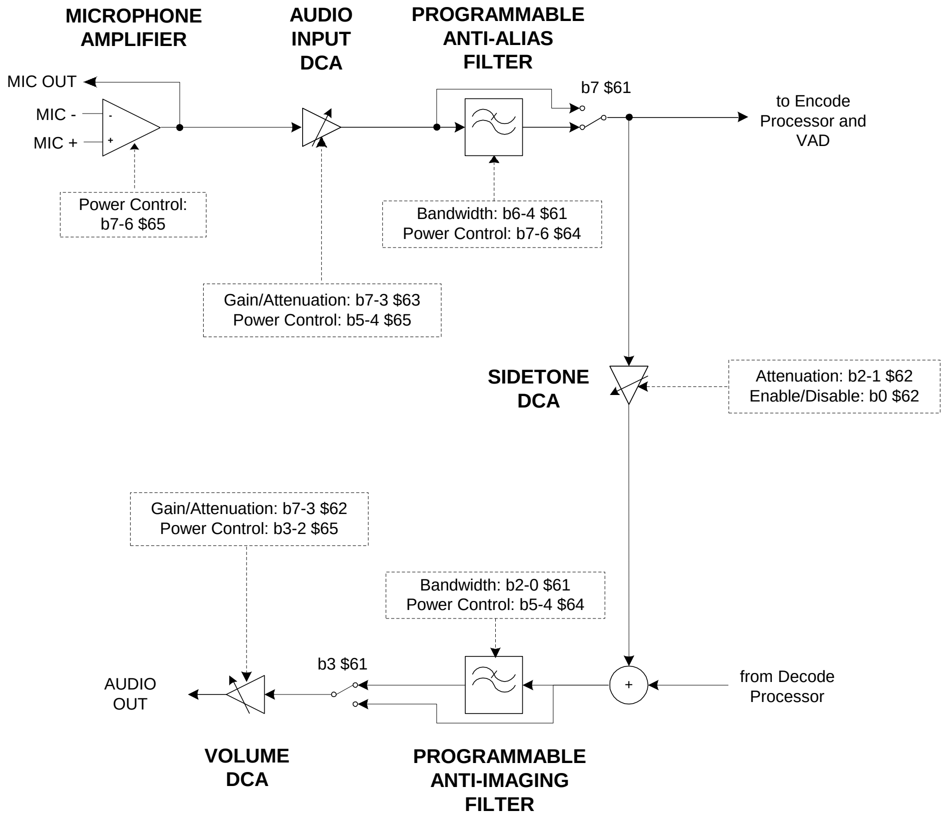

Table 12: Audio Path Components

5.1 Microphone Amplifier

The CMX649 includes a microphone amplifier on board. This amplifier should be connected to an external microphone in accordance with the microphone manufacturer’s recommendation.

Power control for the CMX649 microphone amplifier is provided through bits 7-6 of the POWER CONTROL 2 ($65) register.

A possible configuration for an electret condenser microphone is provided below. Please note that:

- The suggested component values provide 0dB of gain. Many microphones require significant gain (e.g. 20dB) and will require adjustments in component values.
- Any adjustment to resistor values for gain purposes should be performed while considering the corresponding impact on microphone frequency response.
- Values for R5 and R6 are determined by the microphone impedance.
- Optimal values for the microphone amplifier components are application dependent:

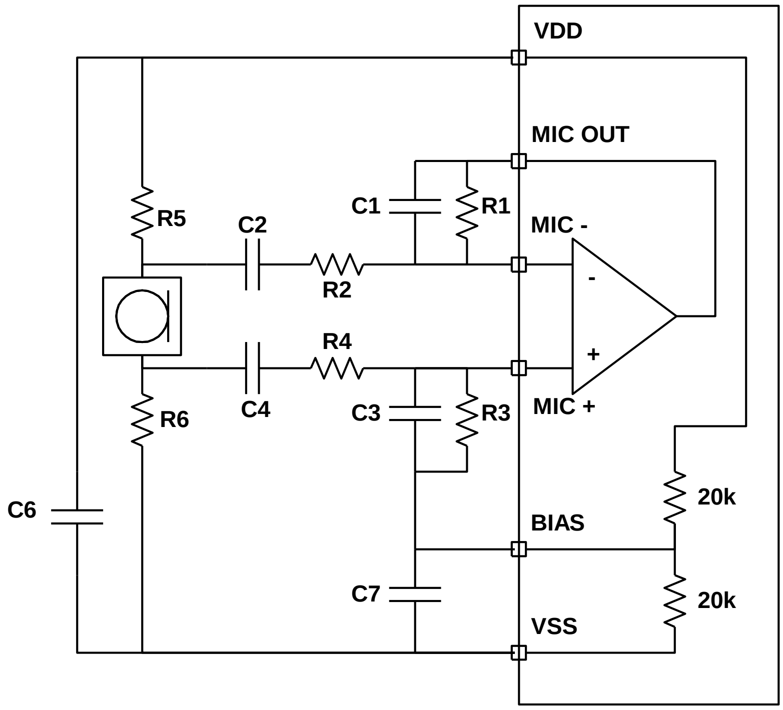

Figure 9: Example of Microphone Connection to CMX649

R1, R3	100kΩ	±10%	R5	100kΩ	±10%
R2, R4	100kΩ	±10%	R6	100kΩ	±10%
C1, C3	100 pF	±20%	C6	1.0μF	±20%
C2, C4	0.01μF	±20%	C7	1.0μF	±20%

Figure 10: Recommended Values for Example Microphone Connection

5.2 Audio Input Digitally Controlled Amplifier (DCA)

The audio input DCA provides gain/attenuation of the microphone amplifier output. The output of the audio input DCA is directed to the programmable anti-alias filter.

The audio input DCA is adjustable in 0.5dB steps over a +7.5dB to –7.5dB range. The desired setting for the audio input DCA is programmed into bits 7-3 of the AUDIO INPUT LEVEL CONTROL ($63) register. Power control for the audio input DCA is provided through bits 5-4 of the POWER CONTROL 2 ($65) register.

5.3 Programmable Anti-Alias Filter (AAF)

The anti-alias filter (AAF) reduces high-frequency content in the input signal, thereby reducing the likelihood that high frequency signals will alias back into the voice band. AAF bandwidth is programmable to accommodate different audio signal bandwidths.

The AAF bandwidth is adjustable between 2.9kHz and 14kHz and is programmed with bits 6-4 of the AAF/AIF BANDWIDTH ($61) register. Bit 7 of this same register ($61) must be cleared to include the AAF in the signal path.

The bandwidth programmed by bits 6-4 of $61 is based upon a switched capacitor filter (SCF) clock frequency of 256kHz. **Any deviation in the SCF clock frequency will cause a proportionate scaling of the AAF bandwidth.**

5.4 Sidetone DCA

The sidetone DCA allows a portion of the audio input to be summed with the recovered receive audio for presentation to the speaker. Sidetone is typically used to minimise the “speaking from the bottom of a well” user experience.

The sidetone DCA is adjustable in 6.0dB steps over a 0dB to –21.0dB range. The desired setting for the sidetone DCA is programmed into bits 2-1 of the VOLUME/SIDETONE LEVEL ($62) register and is enabled with bit 0 of $62.

5.5 Programmable Anti-Image Filter (AIF)

The AIF serves as a reconstruction (i.e. smoothing) filter for the decoder DAC output. AIF bandwidth is programmable to accommodate different audio signal bandwidths.

The AIF bandwidth is adjustable between 2.9kHz and 14kHz and is programmed with bits 2-0 of the AAF/AIF BANDWIDTH ($61) register. Bit 3 of this same register ($61) must be cleared to include the AIF in the signal path.

The bandwidth programmed by bits 2-0 of $61 is based upon a switched capacitor filter (SCF) clock frequency of 256kHz. **Any deviation in the SCF clock frequency will cause a proportionate scaling of the AIF bandwidth.**

5.6 Volume DCA

The volume DCA provides gain/attenuation in 1.5dB steps over a +12.0dB to -33.0dB range. This DCA is capable of driving a 10kΩ load. Since most speakers have impedances in the 4-32Ω range, the volume DCA is not capable of driving a speaker directly, an external speaker amplifier is required.

6 Voice Encoding and Decoding

6.1 General Comments

Regardless of the coding scheme chosen, the CMX649 first converts speech into an ADM representation, and this ADM signal is then converted into the other formats as desired (e.g. linear PCM or non-linear PCM). **Consequently, the device registers related to ADM operation must be properly configured regardless of the chosen coding scheme.**

The CMX649 offers several options for voice coding, including:

- Linear PCM
- A-law PCM
- μ-law PCM
- ADM (which includes CVSD)

The voice coding scheme is common to both encode and decode functions and is discussed in Section 6.4, “Voice Coding Scheme”.

The encoder (decoder) is enabled with bit 7 (bit 3) of the CODEC INTERRUPT CONTROL ($81) register. This register also allows selection of sources for IRQs:

- VAD status changes
- PCM data available or needed for transcoding
- ADM data available or needed for transcoding

A simplified list of encoder/decoder considerations is presented below:

- Coding scheme.
- Input signal source.
- Bit clock rate and source.
- Specific encoder/decoder settings.
- Scrambler/descrambler settings.

6.2 Encoder Specific Considerations

The CMX649 can encode information from three different input sources:

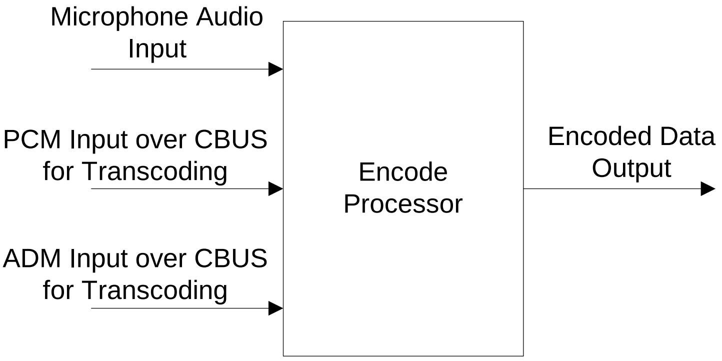
Microphone Audio Input
PCM Input over CBUS for Transcoding
ADM Input over CBUS for Transcoding
Encode Processor
Encoded Data Output

Figure 11: Encode Input Options

The normal mode of CMX649 encoder operation is to encode a microphone input signal. Transcoding is discussed in Section 7, “Transcoding”.

The encode bit clock must be configured in accordance with Section 1.3.1, “Encode Bit Clock”.

The CMX649 registers associated with encoding are presented below:

Function	Controlling Register	Register Bits
Write-only Registers:		
Encode DAC (analogue section) Power Control	$64	3-2
Voice Coding Scheme	$70	2-0
Scrambler Settings	$71	15-13, 9-0
Encode Bit Clock Settings	$72	13, 7-6, 2-0
Encode Bit Clock Source	$73	5-4
Enabling of Encoder and Selection of IRQ Sources	$81	7-4
Encode Mode Settings	$E0	15-8, 0
	$E1	15-0
Idle Channel Enhancement Adjustment	$E3	15-0
Storage Location for PCM Input Data (for transcoding)	$E7	15-0
Storage Location for ADM Input Data (for transcoding)	$E8	7-0
Read-only Registers:		
Encoder Status	$80	7-4
Measured Offset Level	$E5	15-0
Linear PCM Representation of Encoded ADM Signal	$E6	15-0
Encoded ADM Output Data (to be read over C-BUS)	$EA	7-0

Table 13: Registers Associated with Encoding

A detailed block diagram of the encode processor, complete with bit descriptions for signal routing, is provided below:

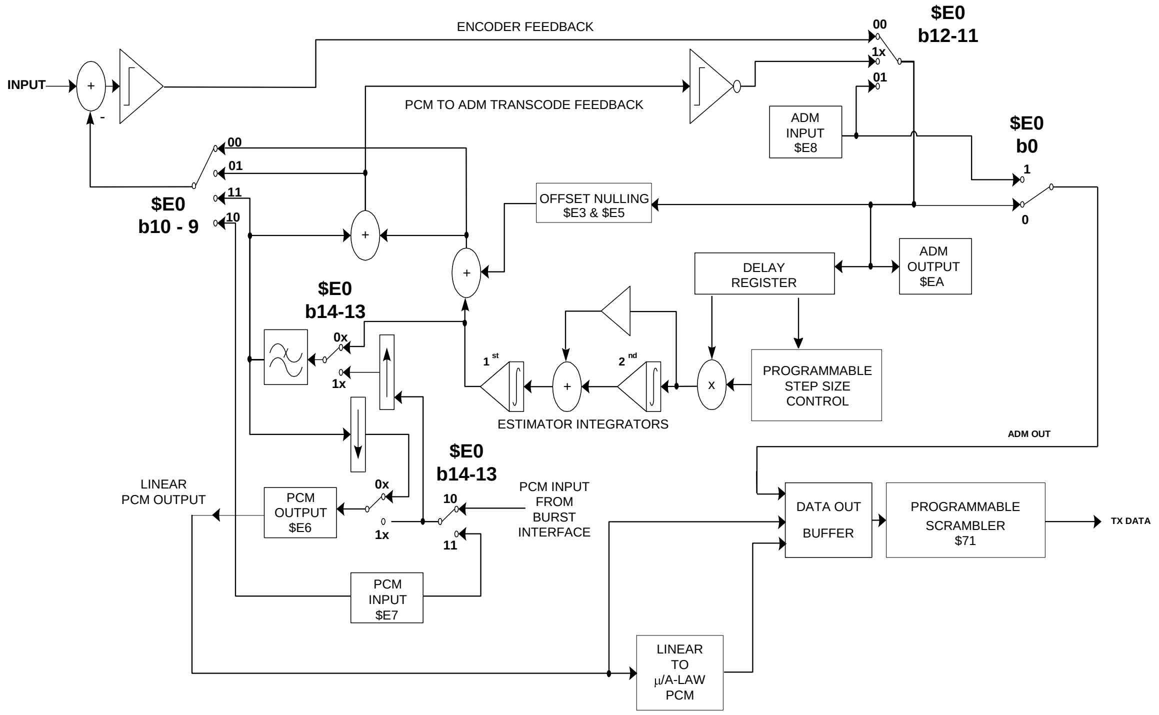

Figure 12: Encoder with Switches Mapped to Bit Positions

6.3 Decoder Specific Considerations

The CMX649 can decode information from three different input sources:

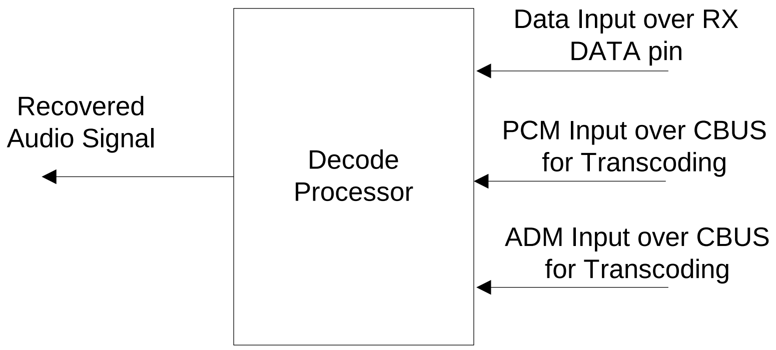
Data Input over RX DATA pin
Recovered Audio Signal
PCM Input over CBUS for Transcoding
Decode Processor
ADM Input over CBUS for Transcoding

Figure 13: Decode Input Options

The normal mode of CMX649 decoder operation is to decode an input data stream to recover the transmitted audio signal. Transcoding is discussed in Section 7, "Transcoding".

The decode bit clock must be configured in accordance with Section 1.4, "Decode Bit Clock".

The CMX649 registers associated with decoding are presented below:

Function	Controlling Register	Register Bits
Write-only Registers:		
Decode DAC (analogue section) Power Control	$64	1-0
Voice Coding Scheme	$70	2-0
Descrambler Settings	$71	12-0
Decode Bit Clock Settings	$72	14, 7-6, 5-3
Decode Bit Clock Source	$73	13-12, 7-6
Data Filter and Slicer Settings	$73	3-0
Enabling of Decoder and Selection of IRQ Sources	$81	3-0
Decode Mode Settings	$D0	15-8
	$D1	15-0
	$E1	0
Idle Channel Enhancement Adjustment	$D3	15-0
Storage Location for PCM Input Data (for transcoding)	$D7	15-0
Storage Location for ADM Input Data (for transcoding)	$D8	7-0
Read-only Registers:		
Decoder Status	$80	3-0
Measured Offset Level	$D5	15-0
Linear PCM Representation of ADM or non-linear PCM Signal	$D6	15-0
Transcoded ADM Output Data (to be read over C-BUS)	$DA	7-0

Table 14: Registers Associated with Decoding

A detailed block diagram of the decode processor, complete with bit descriptions for signal routing, is provided below:

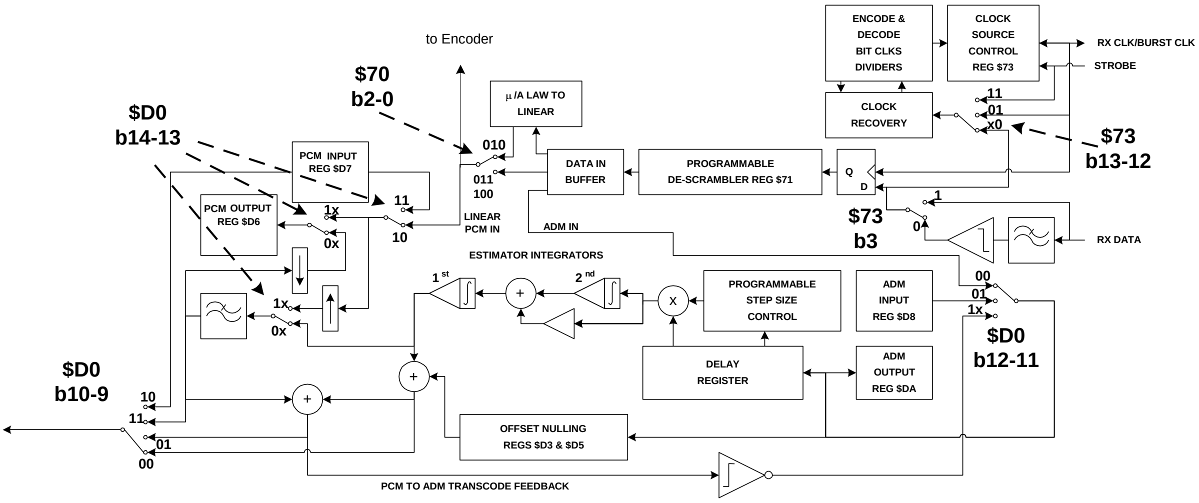

Figure 14: Decoder with Switches Mapped to Bit Positions

6.4 Voice Coding Scheme

The first consideration is the type of coding that is to be performed. The encoding (and decoding) mode is selected by bits 2-0 of the CODEC MODE CONTROL ($70) register.

All CMX649 codec operating modes are burst modes, with the exception of "ADM mode without buffered I/O", which uses a synchronous serial output data stream (i.e. non-burst). Please refer to Section 2.2, "Burst Mode", for more information regarding the burst mode operation.

6.4.1 ADM

Adaptive Delta Modulation, or ADM, uses a variable step height to quantize the difference between the present voice sample and the predicted next voice sample.
Continuously Variable Slope Delta modulation (CVSD), a type of ADM, is commonly used in wireless links for robust voice communications.

The primary coding scheme used by the CMX649 is ADM; all other schemes are developed from the ADM representation of the input signal. **The device registers related to ADM operation must be properly configured regardless of the chosen coding scheme.**

Two ADM modes are possible with the CMX649; burst and non-burst mode. Selection of these modes is made with bits 2-0 of the CODEC MODE CONTROL register.

The performance of the ADM encoder is controlled primarily with two registers; the ENCODER MODE AND SETUP ($E0) register and the ENCODER ADM CONTROL ($E1) register.

Input for the ADM encoder is selected with bits 12-11 of the ENCODER MODE AND SETUP ($E0) register. The input for the ADM encoder is normally provided from the comparator output (please refer to Figure 12).

6.4.1.1 Companding Rule Selection

The comparator uses two signals; the microphone input and the ADM encoder output signal. The ADM encoder output signal can be derived from four different sources but is normally supplied from the ADM estimator output. Bits 10-9 of the ENCODER MODE AND SETUP ($E0) register are responsible for this selection.

Normally, the microphone input signal is compared to its predicted value. The comparator creates a logic one if the microphone input exceeds the predicted value, while a logic zero is created otherwise. The comparator output is then fed to the encoder delay register.

Changes of input signal amplitude that do not exceed the current quantization step size cause the comparator to output alternating ones and zeros. Rapidly changing input signals, however, can exceed the quantization step size and cause the encoder to be in a “slew-rate limited” condition (aka “slope overload”). When this happens, the ADM representation cannot change fast enough to keep up with the microphone input signal, and the comparator produces a string of ones or zeros as a result of the slope overload condition.

The encoder delay register allows from three to six bit times of slope overload to occur before the step size is changed. This time period, also called the “companding rule”, is programmed with bits 9-8 of the DECODE ($D1) and ENCODE ($E1) ADM CONTROL registers.

For example, with the companding rule set to “3 of 3”, a string of three consecutive ones or zeros from the comparator will be detected as slope overload. Once the slope overload condition has been detected, the encoder will adjust the quantization step size so that the ADM representation can more closely track the analogue input signal.

6.4.1.2 Maximum and Minimum Step Height Selection

The maximum and minimum quantization step heights are programmed with bits 12-10 of the ENCODE ($E1) ADM CONTROL register.

Two problems can occur if the max and min step heights are improperly selected; slope overload and granular noise.

Slope overload occurs when the step size is too small; the digital representation cannot adequately track changes in the analogue input. Granular noise occurs when the step size is too large; input signal changes smaller than the step size are not digitised. Granular noise can occur for any step size, so it is advantageous to keep the step size as low as possible while providing adequate protection from slope overload.

Two parameters of concern with any voice coding scheme are the decoded signal level and SINAD. The following considerations describe the impact of step height selection on both of these important parameters.

SINAD Considerations:

1. In general, SINAD drops as input frequency increases. This degradation is due to slope overload, which is maximised with higher input signal amplitude and frequency. Slope overload degrades the SINAD measurement.
2. Offset compensation improves SINAD at mid & upper frequencies, but produces a SINAD hit at low frequencies.
3. For a given step height with higher input frequencies (e.g. 2kHz), SINAD is higher for lower input amplitude than for higher input amplitude.
4. For a large input amplitude at high input frequencies, large max and min step height yields the best SINAD. This is because the larger step heights help counteract slope overload better than small step heights.
5. For a large input amplitude at low input frequencies, large min step yields the best SINAD. Slope overload isn’t as much of a problem with low frequencies because the codec has time to keep up with the signal. With large input signals, there will be fewer instances of threshold effects, so granular noise is reduced. The max step height selection is not critical to SINAD performance in this situation.
6. For a small input amplitude, **small** min step yields best SINAD across all input frequencies. This is because a small step height can easily keep up with input changes without adding too much granular noise. The size of the max step height for small input amplitudes is not critical to SINAD performance.

Decoded Signal Level Considerations:

1. For small input signals, a small minimum step height will maximise the decoder output level for all input frequencies.

2. For large input signals, a large max/min step height will maximise the decoder output level across all input frequencies.

These SINAD and decoded signal level recommendations can be summarized in the following table. These recommendations are general in nature; optimal step height settings are application dependent:

Max and Min Step Heights for Optimal SINAD and Decoded Signal Level Performance		**Input Level**			
		Low (e.g. 50mVrms)		High (e.g. 490mVrms)	
Input Frequency	Low (e.g. 300Hz)	Min	Small	Min	Large
		Max	Not critical	Max	Large
	High (e.g. 2kHz)	Min	Small	Min	Large
		Max	Not critical	Max	Large

Table 15: Summary of Recommendations for Max & Min Step Heights

6.4.1.3 Integrator Time Constant Selection

The syllabic integrator adjusts the quantization step height. The time constant of the syllabic integrator determines how quickly the step height can increase or decrease. The shorter the time constant, the faster the step height can be changed.

The syllabic integrator typically has a time constant that is much longer (e.g. 20x to 30x) than the estimator integrator. For example, for Bluetooth compatibility at 64kbps:

- Syllabic integrator time constant = 16ms
- Estimator integrator time constant = 0.5ms

Typical values for syllabic integrator time constants are in the range of 5-10ms, but optimal values are application dependent.

The time constants for the syllabic and estimator integrators are selected in the ENCODE ($E1) ADM CONTROL register.

6.4.1.4 Second Order Integration

ADM uses an estimator integrator in the encoder feedback path. While this approach yields good voice quality, additional improvement in voice quality can be achieved by adding an additional integration stage to the estimator integrator. This process, known as “second order integration”, introduces a second integrator in the estimator integration to create a smoother reconstructed signal at the decoder.

Second order integration is enabled or disabled with bits 4-3 of the DECODE ($D1) and ENCODE ($E1) ADM CONTROL registers.

When second order integration is used, the encoder can experience instability and oscillations. A “zero” can be added to the estimator integrator transfer function to improve stability with second order integration. The settings for zero selection are contained within bits 2-1 of the ENCODE ($E1) ADM CONTROL register. For example, with a 64kbps data rate:

Encode ADM Control Register ($E1)		**Zero Frequency for 64kbps Data Rate**
Bit 2	Bit 1	
0	0	N/A (first order estimator)
0	1	42.7kHz
1	0	25.6kHz
1	1	14.2kHz

Table 16: Example of Zero Frequency Determination

NOTE: second order integration should not be used for transcoding operation.

6.4.1.5 CVSD

Continuously variable slope delta (CVSD) modulation is a form of ADM. With CVSD, information relating to the slope of the input signal is transmitted to the receiver.

CVSD is achieved whenever ADM coding is performed without second-order integration. (When a second integration is performed on the sampled pulses from the comparator, the resulting signal no longer contains information about the slope of the input signal. Since this signal contains no information about the input signal slope, the coding scheme is no longer CVSD, but purely ADM.)

When CVSD operation is desired the CMX649 should be configured as for ADM operation, but the second-order integration (b4-3 of $D1 and $E1) should be disabled.

6.4.2 Linear PCM

The ADM encoder creates an ADM representation of the microphone input signal, which is then decimated and filtered to create a 13-bit linear PCM representation of the microphone input. (Note that a 16-bit output word is presented, but the lowest three bits of this 16-bit signal should be ignored.)

The CMX649 uses an ADC with 2^{13}=8192 quantization levels and a 2.5V full scale voltage. These values translate to a quantization step voltage of:

$$\frac{2.5V}{8192\,levels} = 305\mu V\,/\,\text{quantization level}$$

The encode bit clock is set up in accordance with Section 1.3.1.2 “Encode Bit Clock from XTAL/CLK Input”.

The decimation rate is determined with bit 15 of the ENCODER MODE AND SETUP ($E0) register. Normal linear PCM operation uses a decimation rate of eight. The relationship between bit clock rate, desired PCM output rate, and decimation rate is as follows:

Bit Clock Rate = Desired PCM Rate x PCM Filter Decimation Rate

The PCM output data can be read out in two different ways:

1. Over the burst mode interface (normal operation):
    a. Data is read out of the TX DATA pin on the falling edge of the BURST CLOCK signal.
        i. Tx data transitions on the BURST CLOCK rising edge
        ii. Tx data is valid on the BURST CLOCK falling edge.
        iii. “Sync” pulses are externally applied to the STROBE pin to signal readiness to accept encoded data.
2. Over the C-BUS interface from the ENCODE LINEAR PCM OUTPUT READ ($E6) register (alternative operation).

6.4.2.1 Suggestions For Linear PCM Operation

While other linear PCM operating modes are possible, the normal linear PCM operating mode involves:

- Encoding of microphone input.
- ADM data filtered and decimated (8x) to achieve linear PCM representation.
- Encoded data is presented on the burst mode interface.
    - TX CLK and RX CLK pins become inputs.
    - BURST CLOCK signal is applied to the RX CLK pin.
    - STROBE pin receives “sync” signal from host microcontroller to frame PCM output data.
    - PCM output data is presented on TX DATA pin.

The following register descriptions illustrate one example of how linear PCM operation can be performed with the CMX649. Bold bit positions indicate encode-relevant bits, while "x" denotes "don't care" bits. Assumptions for these settings include:

- 4.096MHz XTAL/CLK input
- Encode data rate = 64kbps
- Decimation rate = 8
- Output data rate = 8ksps
- Host application provides BURST CLOCK and STROBE timing signals.

These register settings do not include initialisation of other device functions, such as the audio path. Please ensure that all desired functions are properly configured in accordance with the CMX649 data sheet:

POWER CONTROL ($64) register

b7	b6	b5	b4	b3	b2	b1	b0
x	x	x	x	**0**	**1**	x	x

Encoder DAC is enabled and set to lowest power consumption setting.

CODEC MODE CONTROL ($70) register

b7	b6	b5	b4	b3	b2	b1	b0
0	0	0	0	0	**0**	**1**	**0**

Linear PCM with buffered I/O coding scheme selected.

CLOCK DIVIDER CONTROL ($72) register

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
1	x	**1**	0	0	1	0	0	**0**	**0**	x	x	x	**0**	**0**	**0**

SCF prescaler enabled.
Encode bit clock enabled.
SCF filter clock prescaler divider ratio = 1.
SCF filter clock divider ratio = 8.
Encode bit clock prescaler divider ratio = 1.
Encode bit clock divider ratio = 1.

CLOCK SOURCE CONTROL ($73) register

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
0	0	x	x	0	0	0	0	x	x	**0**	**1**	x	x	x	x

Encode bit clock internally generated from XTAL/CLK input.

CODEC INTERRUPT CONTROL ($81) register

B7	B6	B5	B4	B3	B2	B1	B0
1	**x**	**1**	**x**	x	x	x	x

Encoder is enabled and will generate periodic IRQs to indicate PCM data is available.

ENCODER MODE AND SETUP ($E0) register

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
0	**0**	**x**	**0**	**0**	**0**	**0**	**1**	**1**	**1**	**0**	**1**	**1**	**0**	**0**	**0**

PCM filter set to decimation rate of 8.
PCM filter decimates ADM estimator output.
ADM encoder input from comparator.
ADM estimator output drives local decoder output.
Analogue signal offsets are automatically compensated.
Encode VAD decay time constant = 128ms.
Encode VAD attack time constant = 4ms.
Encode VAD output source taken from VAD circuit (normal operation).
ADM output selected for normal operation.

ENCODE ADM CONTROL ($E1) register

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
0	**0**	**0**	**0**	**1**	**0**	**0**	**1**	**0**	**1**	**1**	**0**	**1**	**1**	**0**	**0**

Syllabic time constant set to 512/(3*Bit Rate) = 2.7ms.
Step size set to 5120 (max) and 20 (min).
Companding rule set to 4 of 4.
Estimator time constant set to 48/(3*Bit Rate) = 0.25ms (i.e. 637Hz).
Second order integration enabled with time constant = 0.125ms (i.e. 1273Hz).
Zero inserted at 2.5/Bit Rate = 0.039ms (i.e. 4074Hz).
No zero inserted at half the bit rate.

6.4.3 Non-Linear PCM

Non-linear PCM applies a non-linear quantization characteristic to the sampled voice signal. This technique improves dynamic range by reducing the quantization step height for small amplitude input signals while increasing the step height for large amplitude input signals.

The two most common non-linear schemes, A-law PCM and μ-law PCM, employ a quantization scheme where the output signal is proportional to the logarithm of the input signal. A-law PCM and μ-law PCM are both supported in the CMX649.

6.4.3.1 A-Law PCM

The CMX649 creates an A-law PCM signal by applying a suitable logarithmic compressor (μ=100) to the linear PCM representation of the microphone input. The resulting 8-bit word is read out of the TX DATA pin on the BURST CLOCK falling edge. (The Tx data transitions on the BURST CLOCK rising edge and is valid on the BURST CLOCK falling edge.) SYNC pulses are externally applied to the STROBE pin to signal readiness to accept encoded data.

Please refer to Section 2.2, “Burst Mode”, for more information on this operating mode.

A-law PCM mode is selected by assigning “100” to bits 2-0 of the CODEC MODE CONTROL ($70) register.

6.4.3.2 μ-Law PCM

The CMX649 creates a μ-law PCM signal by applying a suitable logarithmic compressor (μ=255) to the linear PCM representation of the microphone input. The resulting 8-bit word is read out of the TX DATA pin on the BURST CLOCK falling edge. (The Tx data transitions on the BURST CLOCK rising edge and is valid on the BURST CLOCK falling edge.) SYNC pulses are externally applied to the STROBE pin to signal readiness to accept encoded data.

Please refer to Section 2.2, “Burst Mode”, for more information on this operating mode.

μ-law PCM mode is selected by assigning “011” to bits 2-0 of the CODEC MODE CONTROL ($70) register.

7 Transcoding

The CMX649 provides the ability to convert between different types of voice coding signals. This process, known as transcoding, is available for the following conversions:

1. ADM to:
    a. Linear PCM
    b. A-law PCM
    c. μ-law PCM
2. Linear PCM to:
    a. ADM
    b. A-law PCM
    c. μ-law PCM

At a top level, transcoding can be viewed as follows:

ADM ⇒ **Decimation** ⇒ PCM

PCM ⇒ **Interpolation** ⇒ ADM

Transcoding can be performed by both the encode and decode processors, and the input and output locations for data vary depending on the location of the transcoding. Please note that the information presented in the following tables depend on the correct configuration of the various switching paths within the CMX649 :

Encoder: Transcoding Method	**Locations of Input and Output Data**	
	Input	Output
ADM to Linear PCM	• $E8	• $E6 • TX DATA pin
ADM to μ-law/A-law PCM	• $E8	• TX DATA pin
Linear PCM to ADM	• $E7 • RX DATA pin	• $EA • TX DATA pin
Linear PCM to μ-law/A-law PCM	• $E7 • RX DATA pin	• TX DATA pin

Table 17: Encoder Transcoding Options

Note: the selected ADM encoding mode determines how the transcoded data is provided on the TX DATA pin (e.g. burst interface or non-burst interface). Please review Section 2.2, “Burst Mode”, for more information.

Decoder: Transcoding Method	**Locations of Input and Output Data**	
	Input	Output
ADM to Linear PCM	• $D8 • RX DATA pin	• $D6
Linear PCM to ADM	• $D7 • RX DATA pin	• $DA • Signal available for audio output
μ-law/A-law PCM to ADM	• RX DATA pin	• $DA • Signal available for audio output
μ-law/A-law PCM to Linear PCM	• RX DATA pin	• $D6

Table 18: Decoder Transcoding Options

Depending on the location of the transcoding, the encoder or decoder must be properly configured for transcoding operation. The general plan of attack for transcoding is:

1. Select the desired voice coding scheme in register $70. This selection will determine the format for the output data signal, and therefore, determines the type of transcoding to be performed.
2. Configure the SCF clock and encode/decode bit clocks in registers $72 and $73.
3. Power down the sections of the CMX649 that aren’t required for the transcoding operation. Possible areas for consideration are:
    a. Encoder/decoder DACs
    b. AAF

   c. AIF
   d. Microphone amplifier
   e. Audio amplifier
   f. Volume amplifier
   g. Sidetone amplifier
4. Configure the encoder/decoder as required for the particular mode of transcoding.
5. Enable the encoder/decoder and select the desired IRQ source in register $81.

Second order integration should not be used when transcoding is performed.

7.1 Transcoding with ADM Input

In this operating mode, an ADM input stream is converted to a PCM output signal.

The relationship between desired PCM output rate, ADM input rate, and PCM decimation rate is as follows:

PCM Output Sample Rate = ADM Input Rate ÷ PCM Filter Decimation Rate

The encode/decode bit clock must be programmed to the ADM input rate.

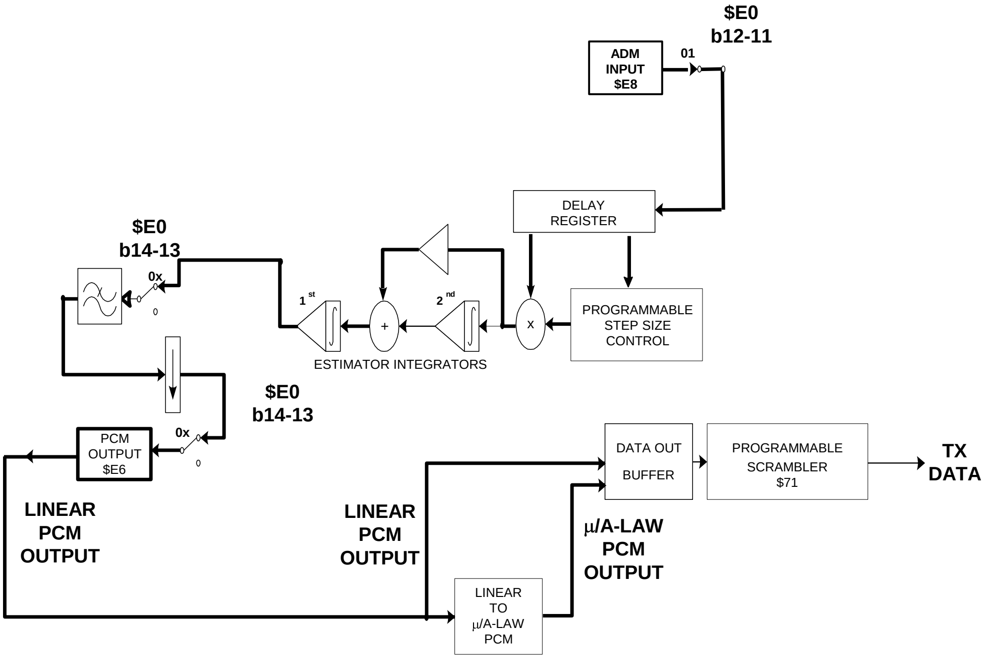

Figure 15: Example of ADM-to-PCM Transcoding with Encoder

The following register description illustrates one example of how ADM-to-PCM transcoding can be performed with the CMX649 encoder. (Transcoding can be performed with the decoder as well.) Other methods may be possible. Bold bit positions indicate transcoding-relevant bits, while “x” denotes “don’t care” bits. Assumptions for these settings include:

- 4.096MHz XTAL/CLK input.
- Input ADM data is written, over the C-BUS interface at 64kbps, into the ENCODE ADM INPUT TEST ($E8) register.
- Encode bit clock = 64kbps, generated internally from XTAL/CLK pin input.
- PCM filter decimation rate = 8.
- Output data rate = 8ksps at 16bits/sample (Linear PCM).
- PCM output samples are read out over the C-BUS interface from the ENCODE LINEAR PCM OUTPUT READ ($E6) register.

These register settings do not include initialisation of other device functions (i.e. audio path. Please ensure that all desired functions are properly configured in accordance with the CMX649 data sheet:

CODEC MODE CONTROL ($70) register							
b7	b6	b5	b4	b3	b2	b1	b0
0	0	0	0	0	**0**	**1**	**0**
Linear PCM with buffered I/O coding scheme selected.							

CLOCK DIVIDER CONTROL ($72) register															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
1	x	**1**	0	0	1	0	0	**0**	**0**	x	x	x	**0**	**0**	**0**
SCF prescaler enabled. Encode bit clock enabled. SCF filter clock prescaler divider ratio = 1. SCF filter clock divider ratio = 8. Encode bit clock prescaler divider ratio = 1. Encode bit clock divider ratio = 1.															

CLOCK SOURCE CONTROL ($73) register															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
0	0	x	x	0	0	0	0	x	x	**0**	**1**	x	x	x	x
Encode bit clock internally generated from XTAL/CLK input.															

CODEC INTERRUPT CONTROL ($81) register							
B7	B6	B5	B4	B3	B2	B1	B0
1	**x**	**1**	**x**	x	x	x	x
Encoder is enabled and will generate periodic IRQs to indicate PCM data is available.							

ENCODER MODE AND SETUP ($E0) register															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
0	**0**	**x**	**0**	**1**	**x**	**x**	**0**	**x**	**x**	**x**	**x**	**x**	**x**	**x**	**0**
PCM filter set to decimation rate of 8. PCM filter decimates ADM estimator output. ADM encoder gets input from C-BUS register $E8. ADM output selected for normal operation.															

ENCODE ADM CONTROL ($E1) register															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
0	**0**	**0**	**0**	**1**	**0**	**0**	**1**	**0**	**1**	**1**	**0**	**0**	**0**	**0**	**0**
Syllabic time constant set to 512/(3*Bit Rate) = 2.7ms. Step size set to 5120 (max) and 20 (min). Companding rule set to 4 of 4. Estimator time constant set to 48/(3*Bit Rate) = 0.25ms. No second order integration. No zero inserted at half the bit rate.															

The above scenario can easily be modified for μ-law or A-law PCM:

- Adjust CODEC MODE CONTROL ($70) register for the desired scheme.
- Output data must be read over burst interface for μ-law or A-law PCM.
  - Output data is read out of TX DATA pin.
  - Output data is supplied at 8 bits per sample for μ-law or A-law PCM.
  - Host microcontroller must apply “sync” pulses to the STROBE pin and a BURST CLOCK signal to the RX CLK pin in accordance with Section 2.2, “Burst Mode”.

7.2 Transcoding with PCM Input

In this operating mode, a PCM input stream is converted to an ADM output signal.

The relationship between desired ADM output rate, PCM input sample rate, and PCM filter interpolation rate is as follows:

ADM Output Data Rate = PCM Input Sample Rate x PCM Filter Interpolation Rate

The encode/decode bit clock must be programmed to the desired ADM output data rate.

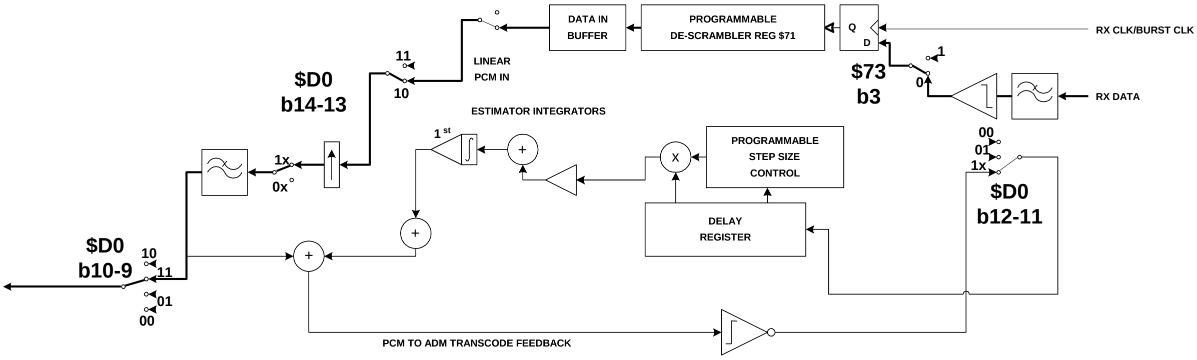

Figure 16: Example of PCM-to-ADM Transcoding in Decoder

The following register description illustrates one example of how PCM-to-ADM transcoding can be performed with the CMX649 decoder. (Transcoding can be performed with the encoder as well.) Other methods may be possible. Bold bit positions indicate transcoding-relevant bits, while “x” denotes “don’t care” bits. Assumptions for these settings include:

- 4.096MHz XTAL/CLK input.
- Input linear PCM data at 8ksps (16 bits per sample) is supplied to RX DATA pin.
  - BURST CLOCK signal (8ksps x 16bits/sample =128kHz) is applied to RX CLK pin to clock in the PCM input data.
  - SYNC pulses are applied to STROBE pin at 8kHz rate to mark PCM byte boundaries.
- Decode bit clock = 64kbps, generated internally from CLK pin input.
- PCM filter interpolation rate = 8.
- Output data rate = 64kbps.
- ADM output signal is applied to the audio path for presentation to an external speaker.

These register settings do not include initialisation of other device functions (i.e. audio path). Additionally, this scenario allows for the decoder DAC to be disabled, but power control register settings are not shown in this example. Please ensure that all desired functions are properly configured in accordance with the CMX649 data sheet:

CODEC MODE CONTROL ($70) register							
b7	b6	b5	b4	b3	b2	b1	b0
0	0	0	0	0	**0**	**1**	**0**
Linear PCM with buffered I/O coding scheme selected.							

CLOCK DIVIDER CONTROL ($72) register															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
1	**1**	x	0	0	1	0	0	**0**	**0**	**0**	**0**	**0**	x	x	x
SCF prescaler enabled. Decode bit clock enabled. SCF filter clock prescaler divider ratio = 1. SCF filter clock divider ratio = 8. Decode bit clock prescaler divider ratio = 1. Decode bit clock divider ratio = 1.															

CLOCK SOURCE CONTROL ($73) register															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
0	0	x	x	0	0	0	0	**0**	**1**	x	x	**0**	**1**	**0**	**1**

<table>
<tr><td>Decode bit clock internally generated from XTAL/CLK input.
Data filter not bypassed.
Data filter and slicer power control set for rates between 32kbps-64kbps.
Data filter in wide bandwidth mode.</td></tr>
</table>

<table>
<tr><th colspan="8">CODEC INTERRUPT CONTROL ($81) register</th></tr>
<tr><td>B7</td><td>B6</td><td>B5</td><td>B4</td><td>B3</td><td>B2</td><td>B1</td><td>B0</td></tr>
<tr><td>x</td><td>x</td><td>x</td><td>x</td><td>1</td><td>x</td><td>1</td><td>x</td></tr>
<tr><td colspan="8">Decoder is enabled and will generate periodic IRQs to indicate when PCM data is needed.</td></tr>
</table>

<table>
<tr><th colspan="16">DECODER MODE AND SETUP ($D0) register</th></tr>
<tr><td>b15</td><td>b14</td><td>b13</td><td>b12</td><td>b11</td><td>b10</td><td>b9</td><td>b8</td><td>b7</td><td>b6</td><td>b5</td><td>b4</td><td>b3</td><td>b2</td><td>b1</td><td>b0</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>1</td><td>x</td><td>1</td><td>1</td><td>0</td><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td><td>0</td></tr>
<tr><td colspan="16">PCM filter set to interpolation rate of 8.
PCM filter interpolates linear PCM input from burst mode interface.
ADM decoder gets input from digital feedback.
ADM output selected for “Interpolated PCM Output” mode.</td></tr>
</table>

<table>
<tr><th colspan="16">DECODE ADM CONTROL ($D1) register</th></tr>
<tr><td>b15</td><td>b14</td><td>b13</td><td>b12</td><td>b11</td><td>b10</td><td>b9</td><td>b8</td><td>b7</td><td>b6</td><td>b5</td><td>b4</td><td>b3</td><td>b2</td><td>b1</td><td>b0</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>1</td><td>0</td><td>1</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td colspan="16">Syllabic time constant set to 512/(3*Bit Rate) = 2.7ms.
Step size set to 5120 (max) and 20 (min).
Companding rule set to 4 of 4.
Estimator time constant set to 48/(3*Bit Rate) = 0.25ms.
No second order integration.
No zero inserted at half the bit rate.</td></tr>
</table>

8 Conclusion

The CMX649 is a very flexible voice codec that can successfully serve many different applications. It is hoped that the information presented in this application note will help the reader better understand the many capabilities of the CMX649.

CML does not assume any responsibility for the use of any algorithms, methods or circuitry described. No IPR or circuit patent licenses are implied. CML reserves the right at any time without notice to change the said algorithms, methods and circuitry and this product specification. CML has a policy of testing every product shipped using calibrated test equipment to ensure compliance with this product specification. Specific testing of all circuit parameters is not necessarily performed.

CML Microcircuits (UK) Ltd *COMMUNICATION SEMICONDUCTORS*	**CML Microcircuits (USA) Inc.** *COMMUNICATION SEMICONDUCTORS*	**CML Microcircuits (Singapore) Pte Ltd** *COMMUNICATION SEMICONDUCTORS*
Tel: +44 (0)1621 875500 **Fax:** +44 (0)1621 875600 **Sales:** sales@cmlmicro.com **Tech Support:** techsupport@cmlmicro.com	**Tel:** +1 336 744 5050 800 638 5577 **Fax:** +1 336 744 5054 **Sales:** us.sales@cmlmicro.com **Tech Support:** us.techsupport@cmlmicro.com	**Tel:** +65 62 888129 **Fax:** +65 62 888230 **Sales:** sg.sales@cmlmicro.com **Tech Support:** sg.techsupport@cmlmicro.com
- www.cmlmicro.com -		