DIGITAL VOICE SYSTEMS, INC.

The Speech Compression Specialists

AMBE-4020™ Half-Duplex and AMBE-4020™ Full-Duplex Vocoder Chips

User's Manual

Version 2.6

October, 2021

AMBE-4020™ Vocoder Chip
Users Manual
Version 2.6
October, 2021

(The most up to date version of the manual is always available at www.dvsinc.com)

Digital Voice Systems, Inc.
600 Cleveland Street
STE 795
Clearwater, FL 33755

Digital Voice Systems, Inc.
END USER PRODUCT License Agreement

This non-exclusive END USER PRODUCT License Agreement (EUPLA) is a legal agreement between the customer of this PRODUCT (the END USER) and Digital Voice Systems, Inc. (DVSI) covering the terms and conditions under which this DVSI PRODUCT and DVSI's proprietary content (that may consist of and is not limited to software, hardware, documentation and other material) is licensed to the END USER.

1. Preliminary Statements and Definitions

1.1 By installing, or otherwise using this PRODUCT the END USER agrees to be bound by the terms and conditions set forth in this EUPLA. If the END USER does not agree to the terms and conditions set forth in this EUPLA, then the END USER must not install or use the PRODUCT and shall immediately return the PRODUCT to DVSI as set forth in Section 13.

a) The PRODUCT shall mean the Hardware, Software, Documentation and other materials that were provided by DVSI, either directly or indirectly through distributors or third parties, to the END USER as part of a sale, delivery or other transaction.

b) Hardware can be in the form of Integrated Circuits (such as Digital signal Processors), Circuit boards and electronics enclosed in a chassis. DVSI's Vocoder Chips (AMBE-3000™, AMBE-4020™, etc…) are an example of an Integrated Circuit.

c) Software can be in form of computer code, firmware masked into an IC or stored or embedded into ROM or RAM or Flash memory, or software stored on any media (such as CD-ROM, floppy disk, hard drive, solid-state memory or the Internet)

d) Documentation means written or electronic information, including user manuals, technical documents, training materials, specifications or diagrams, that pertain to or are delivered with the PRODUCT in any manner (including in print, on CD-ROM, or on-line).

1.2 DVSI has developed a number of voice coding methods and algorithms (the "Technology") which include DVSI's Advanced Multi-Band Excitation ("AMBE®"), AMBE+™, and AMBE+2™ voice coders. The Technology codes speech at low bit rates and may include error correction, echo cancellation and other auxiliary functions.

1.3 "DVSI Voice Compression Software" shall mean the voice coding Software that implements or embodies the Technology and is embedded into or otherwise provided with the PRODUCT.

1.4 "DVSI Voice Codec" shall mean the DVSI Voice Compression Software, any PRODUCT Hardware into which the DVSI Voice Compression Software is embedded or executed and any associated Documentation.

1.5 DVSI represents that it owns certain "Proprietary Rights" in the PRODUCT including patent rights, copyrights, trademarks and trade secrets. These rights include one or more of the following US Patents #6,912,495; #7,970,606; #8,036,886; #8,200,497, #8,315,860; #8,359,197; #8,433,562; and #8,595,002; and under other US and foreign patents, related patents and patents pending. AMBE®, AMBE+™ and AMBE+2™ are registered trademarks and trademarks of Digital Voice Systems, Inc.

1.6 "END USER" shall mean the person and/or organization to whom the DVSI Vocoder Product (software or hardware) was delivered or provided to as specified in the purchase order or other documentation. In the event that the END USER transfers his rights under this license to a third party as specified in Section 3.0, then this third party shall become an "END USER".

1.7 DVSI reserves the right to make modifications and other changes to its products and services at any time and to discontinue any product or service without notice.

2. License Granted

2.1 Subject to the conditions herein and upon initial use of the DVSI Product, DVSI hereby grants to END USER a non-exclusive, limited license to use the DVSI Voice Compression Software and Technology solely as included by DVSI within the PRODUCT. No license, either expressed or implied, is granted for any use of the DVSI's Proprietary Rights on any other device or Hardware or in any manner other than within the original unmodified PRODUCT purchased from DVSI. No license is granted to copy or modify the DVSI Voice Compression Software or the PRODUCT either in whole or in part.

2.2 No license, right or interest in any trademark, trade name or service mark of DVSI is granted under this Agreement. END USER acknowledges that the PRODUCT may contain trade secrets of DVSI, including but not limited to the specific design, and associated interface information.

2.3 END USER shall not copy, extract, reverse engineer, disassemble, de-compile or otherwise reduce the DVSI Voice Compression Software to human-readable form. END USER shall not alter, duplicate, make copies of, create derivative works from, distribute, disclose, provide or otherwise make available to others, the DVSI Voice Compression Software and Technology and/or trade secrets contained within the PRODUCT in any form to any third party without the prior written consent of DVSI. The END USER shall implement reasonable security measures to protect such trade secrets.

2.4 This is a license, not a transfer of title, to the DVSI Voice Compression Software, Technology and Documentation, and DVSI retains ownership and title to all copies.

3. Transfer of License

3.1 The END USER shall have the right to transfer the rights under this EUPLA to a third party by providing the third party with a copy of this EUPLA and obtaining the third parties agreement to all the terms and conditions under this EUPLA. In the event END USER does transfer their rights to a third party in accordance with this EUPLA, then the third party transferee shall become the new END USER and all rights under this EUPLA shall terminate with respect to the transferor in accordance with Section 4.

4. Term and Termination

4.1 This Agreement is effective upon initial delivery of the PRODUCT and shall remain in effect until terminated in accordance with this agreement.

4.2 This Agreement shall terminate automatically without notice from DVSI if END USER fails to comply with any of the material terms and conditions herein. END USER may terminate this Agreement at any time upon written notice to DVSI certifying that END USER has complied with the provisions of Section 13.

4.3 Upon termination of this Agreement for any reason, END USER shall: (a) discontinue all use of the PRODUCT; (b) return the PRODUCT and documentation purchased or acquired, or in Licensee's possession, to DVSI; (c) have no further rights to any DVSI Software or the Technology without a separate written license from DVSI; All confidentiality obligations of Customer and all limitations of liability and disclaimers and restrictions of warranty shall survive termination of this Agreement. In addition, the provisions of the sections titled "U.S. Government End User Purchasers" and "General Terms Applicable to the Limited Warranty Statement and End User License" shall survive termination of this Agreement.

5. Payments

5.1 In consideration of the materials delivered as part of the PRODUCT, and in consideration of the license granted by DVSI for the PRODUCT, and in consideration of DVSI's performance of its obligations hereunder, the END USER agrees to pay to DVSI the fees as specified in DVSI's invoice. Payments of fees shall be received by DVSI prior to shipment of the PRODUCT.

6. Proprietary Notices

6.1 END USER shall maintain and not remove or alter any copyright or proprietary notice on or in the PRODUCT.

6.2 Reproduction of non-proprietary information found in DVSI Users Manuals or data sheets is permissible only if the END USER reproduces without alteration, and includes all copyright and other proprietary notices, all associated warranties, conditions and limitations on all copies, in any form.

7. Proprietary Information

7.1 The parties agree that the PRODUCT shall be considered Proprietary Information.

7.2 Except as otherwise provided in this Agreement, END USER shall not use, disclose, make, or have made any copies of the Proprietary Information, in whole or in part, without the prior written consent of DVSI.

8. Limited Warranty

8.1 DVSI warrants the PRODUCT to be free from defects in materials and workmanship under normal use for a period of ninety (90) days from the date of delivery. The date of delivery is set forth on the packaging material in which the Product is shipped. This limited warranty extends only to the Customer who is the original purchaser. If the PRODUCT is found to be defective and the condition is reported to DVSI, within the warranty period, DVSI may, at its option, repair, replace, or refund of the purchase price of the PRODUCT. DVSI may require return of the PRODUCT as a condition to the remedy.

> Restrictions. This warranty does not apply if the PRODUCT (a) has been altered, (b) has not been installed, operated, repaired, or maintained in accordance with instructions supplied by DVSI, (c) has been subjected to abnormal physical or electrical stress, misuse, negligence, or accident;

8.2 Except as stated in Section 8.1, the PRODUCT is provided "as is" without warranty of any kind. DVSI does not warrant, guarantee or make any representations regarding the use, or the results of the use, of the PRODUCT with respect to its correctness, accuracy, reliability, speech quality or otherwise. The entire risk as to the results and performance of the PRODUCT is assumed by the END USER. After expiration of the warranty period, END USER, and not DVSI or its employees, assumes the entire cost of any servicing, repair, replacement, or correction of the PRODUCT.

8.3 DVSI represents that, to the best of its knowledge, it has the right to enter into this Agreement and to grant a license to use the PRODUCT to END USER.

8.4 Except as specifically set forth in this Section 8, DVSI makes no express or implied warranties including, without limitation, the warranties of merchantability or fitness for a particular purpose or arising from a course of dealing, usage or trade practice, with respect to the PRODUCT. Some states do not allow the exclusion of implied warranties, so the above exclusion may not apply to END USER. No oral or written information or advice given by DVSI or its employees shall create a warranty or in any way increase the scope of this warranty and END USER may not rely on any such information or advice. The limited warranties under this Section 8 give END USER specific legal rights, and END USER may have other rights which vary from state to state.

9. Limitation of Liability

9.1 The END USER agrees that the limitations of liability and disclaimers set forth herein will apply regardless of whether the END USER has accepted the product or service delivered by DVSI.

9.2 In no event shall DVSI be liable for any special, incidental, indirect or consequential damages resulting from the use or performance of the PRODUCT whether based on an action in contract, or for applications assistance, or product support, or tort (including negligence) or otherwise (including, without limitation, damages for loss of business revenue, profits, business interruption, and loss of business information or lost or damaged data), even if DVSI or any DVSI representative has been advised of the possibility of such damages.

9.3 Because some states or jurisdictions do not allow the exclusion or limitation of liability for consequential or incidental damages, the above limitations may not apply to END USER.

9.4 DVSI's maximum liability for damages arising under this Agreement shall be limited to 20% (twenty percent) of the fees paid by END USER for the particular PRODUCT that gave rise to the claim or that is the subject matter of, or is directly related to, the cause of action.

10. Taxes

10.1 All payments required under Section 5 or otherwise under this Agreement are exclusive of taxes and END USER agrees to bear and be responsible for the payment of all such taxes (except for taxes based upon DVSI's income) including, but not limited to, all sales, use, rental receipt, personal property or other taxes which may be levied or assessed in connection with this Agreement.

11. Export

11.1 DVSI represents that to the best of its knowledge, this PRODUCT has been shipped from the United States in accordance with the US Export Administration Regulations (EAR). The Purchaser, Distributor, or END USER hereby gives its assurance to DVSI that it will comply with all export and re-export restrictions and regulations of all United States government agencies and authorities as well as when applicable, European Union and other export agencies. Export regulations include and are not limited to restrictions, prohibiting export or re-export of this DVSI PRODUCT (a) into an US-embargoed country or (b) to anyone person or entity named on any of the screening lists of the US Departments of Commerce, State and the Treasury. For a list of parties for which the US Government maintains restrictions on exports, re-exports or transfers of items, reference the Consolidated Screening List (CSL) which may be found at: (https://www.export.gov/article?id=Consolidated-Screening-List). The Purchaser, Distributor, or END USER represents and warrants that they are not located in any such country or named on any such list.

11.2 DVSI takes no responsibility, and will not refund any monies paid, for any shipment that gets delayed, held or seized by US export authorities due to non-compliance with any export law, restriction or regulation. Any Purchaser, Distributor, or END USER of this PRODUCT who is not familiar with US export law should seek professional advice to ensure that they are in full compliance.

11.3 DVSI makes no representation that an export or re-export license is or is not required, and if a license is required, that it will or will not be issued by the U.S. Department of Commerce. The Purchaser, Distributor, or END USER is solely responsible, at their own expense, for obtaining any US or other government permits, licenses or approvals required for the importing and/or exporting the PRODUCT. For up-to-date information regarding United States import / export laws and regulations please visit https://www.export.gov/.

12. Governing Law

12.1 This Agreement is made under and shall be governed by and construed in accordance with the laws of the Commonwealth of Massachusetts, (USA), except that body of law governing conflicts of law. If any provision of this Agreement shall be held unenforceable by a court of competent jurisdiction, that provision shall be enforced to the maximum extent permissible, and the remaining provisions of this Agreement shall remain in full force and effect. This Agreement has been written in the English language, and the parties agree that the English version will govern.

13.0 Notices

13.1 Any notices to DVSI which may be given hereunder shall be sent in writing to: Digital Voice Systems Inc., 600 Cleveland Street STE 795 Clearwater, FL 33755 USA

Table of Contents

List of Figures

List of Tables

1 Product Introduction

1

Digital Voice System's AMBE-4020™ Vocoder Chip is an extremely flexible, low cost, half-duplex, voice compression solution that is ideal for commercial, consumer, and military mobile radio communication applications. As part of DVSI's family of high performance vocoder chips, the AMBE-4020™ Vocoder Chip boasts a built-in 16-bit ADC and 12-bit DAC in a small footprint design that makes it easy to integrate and requires minimal power.

The AMBE-4020™ Vocoder Chip uses proven technology to deliver a level of performance and reliability typically associated only with customized ASICs, but without the associated risks and high development costs. With small quantity, off-the-shelf availability and no licensing fees or royalties, engineers and original equipment manufacturers are able to efficiently design and affordably produce high performance, narrowband, communication equipment.

There are two versions of the AMBE-4020™ Vocoder Chip. The AMBE-4020™ Full-Duplex is a full-duplex version and the AMBE-4020™ is a half-duplex vocoder chip. Both models are similar to each other except a few minor differences. The manual will call out the functions that are unique to each model using colored text. Orange for the AMBE-4020™ Full-Duplex and Green for the AMBE-4020™ Half-Duplex, where the black text is common to both devices. Both models are a compact LQFP package (BGA package available on Special Order).

1.1 Advances in Vocoder Design

The AMBE-4020™ Vocoder Chip implements DVSI's patented AMBE+2™ Voice Compression Algorithm that can operate at virtually any data rate from 2.0 to 9.6 kbps. When enabled, the built-in FEC combines block and convolution codes with optional four-bit soft decision decoding.

With this high degree of flexibility, the user can optimize speech and Forward Error Correction (FEC) rates to provide exceptional robustness to background noise and intelligible speech in degraded channel conditions, even with bit errors (BER) of up to 20%. This level of performance can lead to the successful development and deployment of wireless communication systems in the most demanding environments.

As another member DVSI's family of advanced voice compression products, the AMBE-4020™ has interoperable modes that provide a seamless migration path from systems already using DVSI's AMBE-3000™ and AMBE-2000™ Vocoder Chips.

DVSI Vocoders Interoperable with the AMBE-4020™	
AMBE-2000™ Chip Rates	**YES**
AMBE-3000™ Chip Rates*	**MOST***
MotoTRBO	NO
DMR	NO
dPMR	NO
DSTAR	**YES**
NXDN	NO
APCO Project 25 Half Rate	NO
APCO Project 25 Full Rate	NO
TerreStar	NO
MexSat	NO
GlobalStar	NO
BGAN	NO

Table 1 Vocoder Interoperability

*** NOTE:** The following AMBE-3000™ built-in rates are not supported: 33-37, 47, and 51-55. As a result, MotoTRBO, DMR, dPMR, NXDN and APCO Project 25 Half Rate are supported by the AMBE-3000™, but are not supported by the AMBE-4020™.

1.2 AMBE-4020™ Vocoder Chip Performance

The AMBE-4020™ Vocoder Chip offers designers several convenient features. In Addition to the built-in 16-bit ADC and 12-bit DAC, there is an interface for low-cost digital microphone, automatic Voice/Silence Detection (VAD), adaptive comfort noise insertion (CNI), DTMF and Call Progress Tone detection/regeneration, and low power modes. Additionally, the AMBE-4020™ Vocoder Chip includes serial interfaces for vocoder configuration, status information, as well as, transferring speech and compressed data bits to/from the chip's encoder and decoder. The numerous advanced design features not only ease design constraints, but also reduces additional hardware requirements.

The AMBE-4020™ vocoder chip offers our customers tremendous flexibility in the design of high performance, narrowband, push-to-talk equipment. This small, low-cost platform enables engineers and Original Equipment Manufacturers to efficiently produce mobile radio and other wireless communication solutions where bandwidth is at a premium and high quality voice is crucial to success.

- Superior voice quality, DVSI's latest generation AMBE+2™ Vocoder Technology
- The AMBE-4020™ Full-Duplex supports echo cancellation and echo suppression
- Maximizes channel bandwidth efficiency by supporting data-rates from 2.0 kbps to 9.6 kbps
- User selectable forward error correction rates - 50 bps to 7.2 kbps (total rate not exceeding 9.6 kbps)
- Excellent performance at low data rates and harsh environments
- Robustness to acoustic background noise and channel bit errors
- Advanced features like Noise Suppression, Improved Error Mitigation, and Soft Decision FEC Decoding
- Supports a-law and μ-law companding via I2S interface
- DTMF and single tone detection and regeneration with North American call progress tones

1.3 Design Flexibility/Low Cost Integration

The AMBE-4020™ Vocoder Chip includes a number of advanced features that are combined with low power consumption to offer the affordability, mobility and power efficiency required by virtually all mobile communication devices.

- Complete integrated 16-bit ADC and 12-bit DAC and vocoder in one chip
- Works with most low-cost A/D-D/A codecs
- Input Interfaces available: internal ADC or digital mic or I2S
- Output interfaces available: internal DAC or I2S
- Interface for low-cost digital microphone
- Very low power consumption - Ideal for portable mobile devices
- 80 pin 12mm x 12mm LQFP small package design
- 121 ball 8mm x 8mm BGA package available only on special order
- Push-to-talk signaling
- Half-duplex Operation or Full-duplex Operation
- No licensing fees or royalties
- Off-the-shelf availability for quick delivery

1.4 Proven Technology

The value of DVSI's AMBE® Technology goes beyond low bit rate and voice quality. It has been thoroughly evaluated and tested by international manufacturers under various conditions using a variety of languages. This assures the user is getting the best vocoder available and makes the DVSI vocoder the logical choice for all your voice compression needs. DVSI has been providing Voice Compression technology for more than 25 years. DVSI's technology has been proven a key component in the overall success of communication systems worldwide.

2 Hardware Information

2

The AMBE-4020™ Vocoder Chip provides excellent performance with the latest low power innovations and high precision mixed-signal capability.

2.1 AMBE-4020™ and AMBE-4020™ Full-Duplex Vocoder Chip Markings

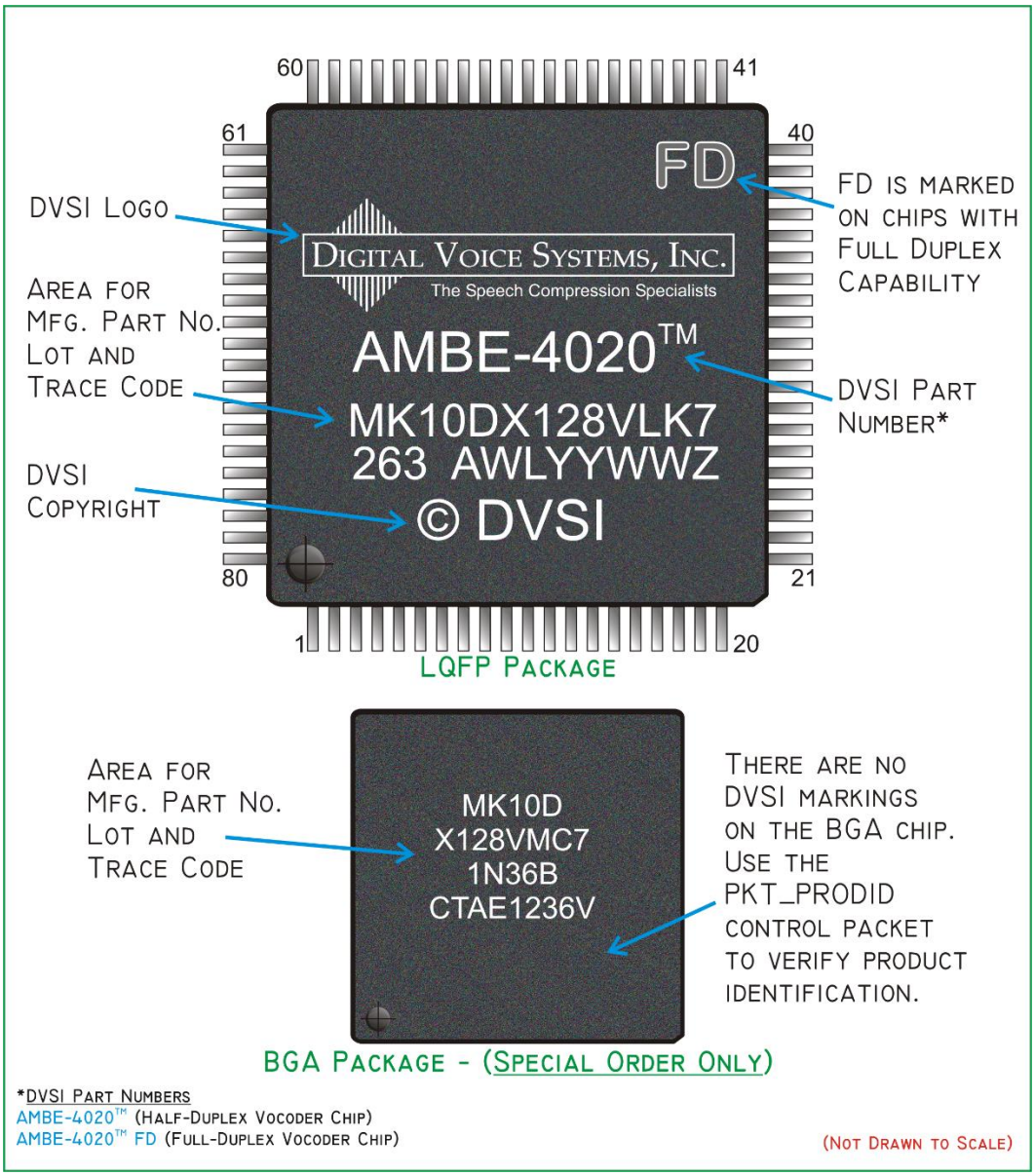


Figure 1 AMBE-4020™ Half-Duplex and Full-Duplex Vocoder Chip Markings

DVSI Logo --- Representation of Digital Voice Systems, Inc. Logo.
DVSI Part Number --- The DVSI device part number is AMBE-4020™
© DVSI --- Copyright Digital Voice Systems, Incorporated

2.2 AMBE-4020™ BGA Packaging labeling

The AMBE-4020™ Vocoder chip is available in a **BGA package only as a Special Order**. When a BGA chip is shipped, to clearly indicate what AMBE-4020™ version is shipped to our customers, there is a verification label included on the tray of the vocoder chip shipment. An example of the AMBE-4020™ Vocoder chip container tray notice for the AMBE-4020™ Full-Duplex (BGA package) is shown in Figure 2.

Figure 2 ABME-4020™ Shipping Tray Label for Special Order BGA chip only.

As an additional measure, the sealed package of AMBE-4020™ BGA vocoder chips will have another notice to indicate the chips were verified to be programed correctly. An example of the AMBE-4020™ Full Duplex (BGA package) notice that is placed on the outside of the package is shown in Figure 3.

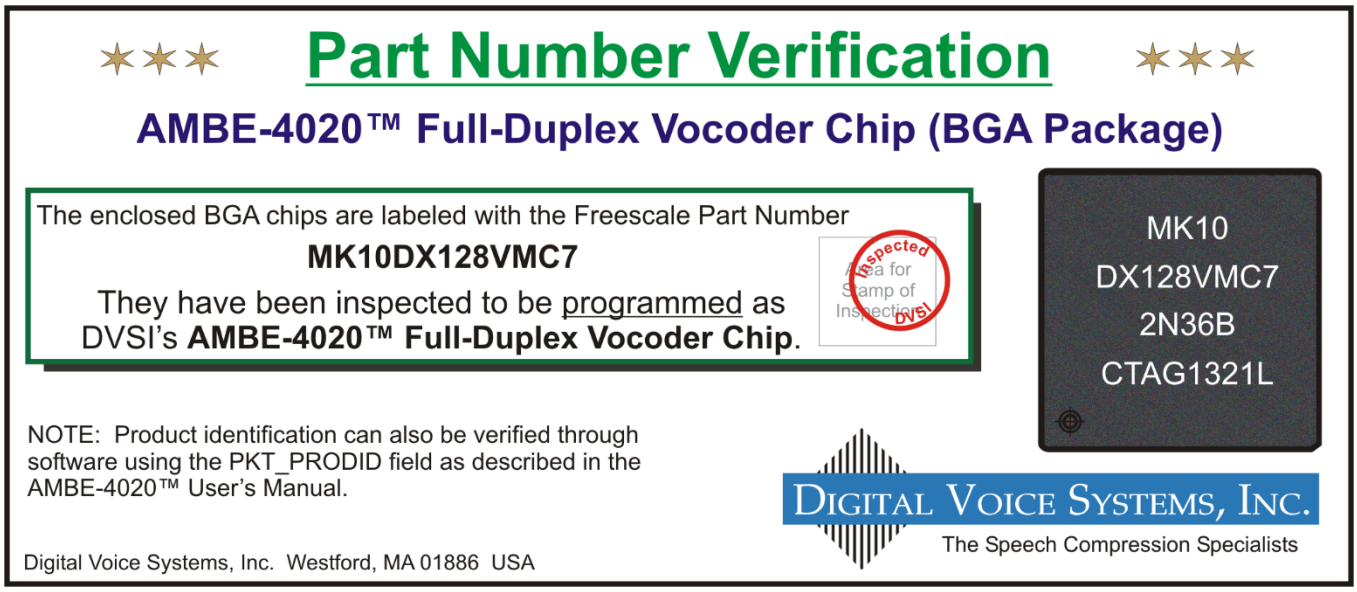


Figure 3 AMBE-4020™ Package Label example for Special Order BGA chip only.

2.3 LQFP Pin Assignments

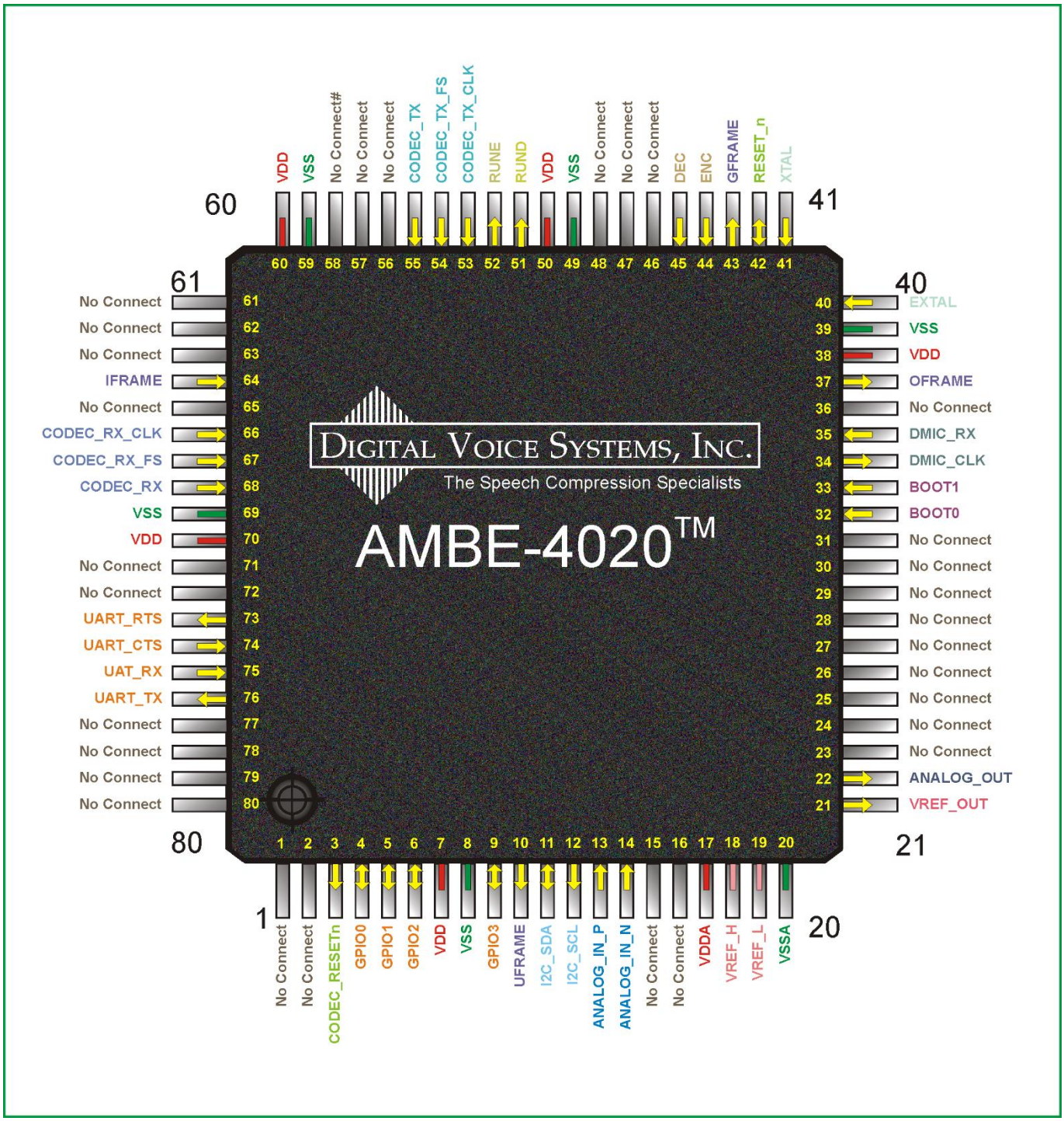


Figure 4 AMBE-4020™ Vocoder Chip Pins for LQFP Package

2.4 BGA Pin Assignments (Special Order Only)

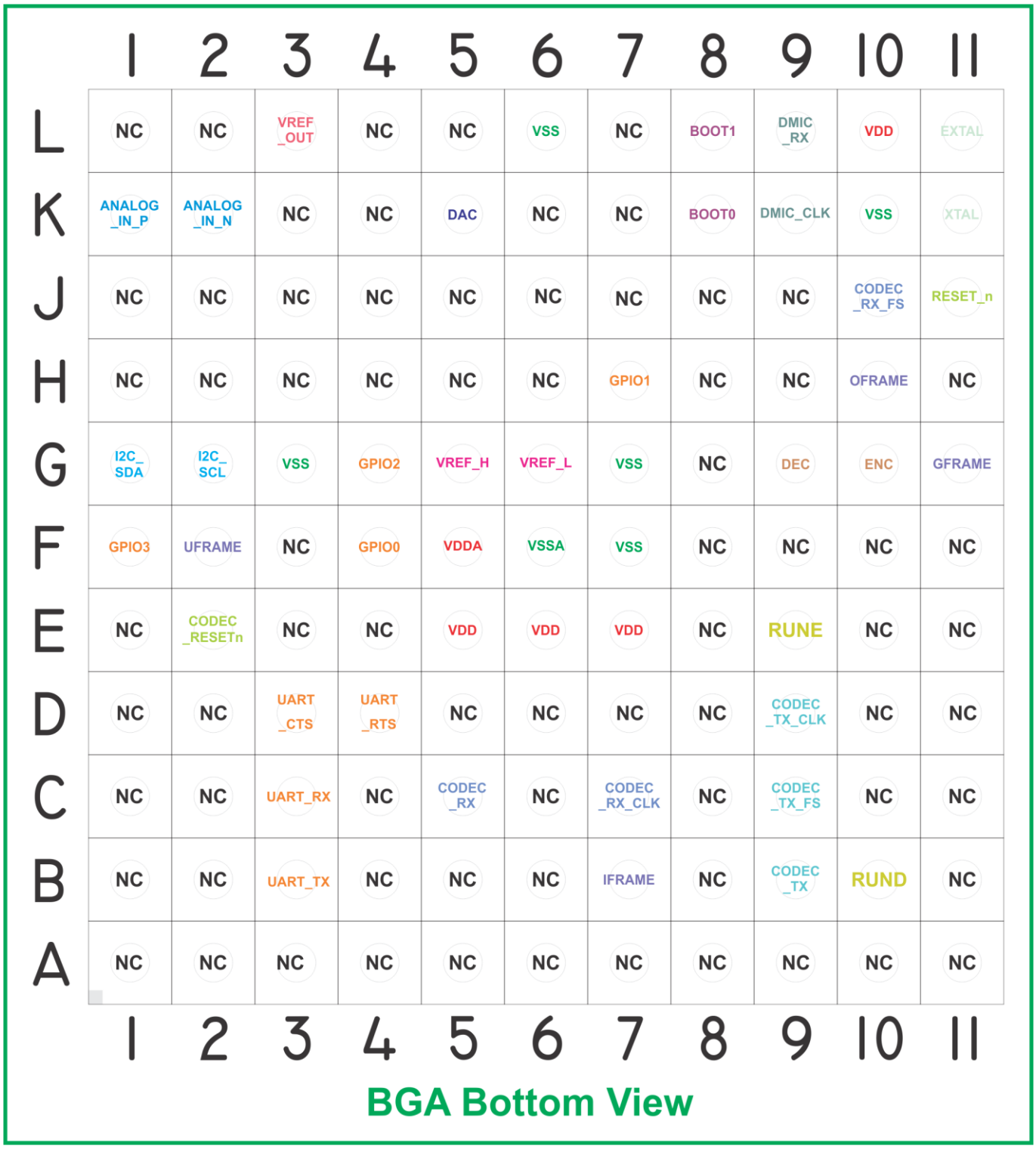


Figure 5 AMBE-4020™ Vocoder Chip Pins for BGA Package (Special Order Only)

2.5 Pin Out Table

Pin Name	Pin Number		Pin Type	Notes
	LQFP	BGA		
CODEC_RESETn	3	E2	Output	Output to Reset an external Codec. This signal is active low.
GPIO0	4	F4	I/O	The GPIO0-GPIO3 pins are controlled using a control packet (PKT_GPIO). The packet allows the direction of each pin to be specified independently. It also allows the state for output pins to be specified and it allows for the state of input pins to be returned in a response packet. The default state after reset is that GPIO0-GPIO3 are disabled.
GPIO1	5	H7	I/O	
GPIO2	6	G4	I/O	
GPIO3	9	F1	I/O	
UFRAME	10	F2	Output	User Frame. This 50 Hz signal is synchronous with OFRAME and may be customized by the user. By default, The signal rises 5 ms after OFRAME rises and falls 5 ms after OFRAME falls, such that it looks like OFRAME delayed by a quarter cycle. The rise time and fall time are configurable in 125 µs increments, such that both the delay and duty cycle can be configured. Another processor might use the signal to perform actions synchronously to the AMBE-4020™. The signal provides two edges that can be configured via a control packet (PKT_UFRAME).
I2C_SDA	11	G1	I/O	This pin is the data pin used to transfer configuration data to an external codec using the I2C protocol. The pin is open drain enabled and must be pulled high.
I2C_SCL	12	G2	I/O	This pin is the clock pin used to transfer configuration data to an external codec using the I2C protocol. The pin is open drain enabled and must be pulled high.
ANALOG_IN_P	13	K1	Input	These pins are the internal ADC's differential input.
ANALOG_IN_N	14	K2	Input	
VDDA	17	F5	Power	Analog supply voltage: This can be connected to the same supply as VDD but should be isolated to minimize noise and error. See Drawing on page 111.
VREFH	18	G5	Input	ADC positive reference voltage: This should be nominally the same voltage as VDDA but should be isolated to minimize noise and error. See Drawing on page 111. The voltage level on this pin is used as a reference for ADC and DAC. Note: this pin should be tied to VDD if the ADC and DAC are not used.
VREFL	19	G6	Input	ADC negative reference voltage: This should be the same voltage as VSS but should be isolated to minimize noise and error. See Drawing on page 111. The level on this pin is used as a reference for ADC and DAC.
VSSA	20	F6	GND	Analog Ground
VREF_OUT	21	L3	Output	No connection required. Internally-generated Voltage Reference output. When the ADC and/or DAC are active, the nominal voltage level on this pin is 1.195V.
ANALOG_OUT	22	K5	Output	Analog Output from the internal DAC.
BOOT0	32	K8	Input	

BOOT1	33	L8	Input	BOOT0 and BOOT1 are used to select one of four boot configurations. Boot Configuration 0 is hard-coded whereas Boot Configurations 1-3 are stored in persistent memory and may be customized using PKT_BOOTCFG.
DMIC_CLK	34	K9	Output	When the Digital Mic is selected as the codec mode input speech source, the AMBE-4020™ outputs a 1.92 MHz clock on DMIC_CLK.
DMIC_RX	35	L9	Input	The AMBE-4020™ receives PDM speech data from a digital microphone. See Schematic on Page 111
OFRAME	37	H10	Output	A 50 Hz framing signal. In codec mode, the AMBE-4020™ operates using 20 ms frames. The frame interval is the period between two successive rising edges of OFRAME. When the encoder is running, it outputs packets synchronously with OFRAME. The onset of packet transmission occurs between 134 µs and 144 µs after the falling edge of OFRAME. The falling edge of OFRAME also serves as the deadline for receiving a packet to be decoded. If no packets are available to decode at this deadline, then the chip will automatically insert a "repeat" frame that has small impact on voice quality when the frequency of repeats is low. The signal has a duty cycle of 50% ± 2.5%.
EXTAL	40	L11	Input	Use EXTAL/XTAL to connect a 4 Mhz Crystal/Clock input as shown in Section 2.15: Crystal / Oscillator Usage.
XTAL	41	K11	Input	
RESETn	42	J11	I/O	A LOW on this pin indicates that the AMBE-4020™ is in a Reset state. This pin is open drain and has an internal pull-up device. Asserting RESET wakes the device from any mode.
GFRAME	43	G11	Output	The AMBE-4020™ can generate a 50 Hz FRAME signal on this pin. The signal could be used as a source for the IFRAME input. The signal could also be used as the source of a 50 Hz clock in packet mode. The packet field PKT_GDIV is used to enable the clock and specify its frequency. The clock is disabled during sleep/halt modes. The duty cycle is 50%.
ENC	44	G10	Input	In push-to-talk codec mode when ENC is high the encoder runs and the AMBE-4020™ outputs a packet every 20 ms. When ENC is low but DEC is high the decoder runs and expects to receive a channel packet every 20 ms. When both ENC and DEC are low, the chip conserves power as determined by PKT_PMODE. Since push-to-talk codec mode only supports half-duplex operation, if both ENC and DEC are high, only the encoder runs.
DEC	45	G9	Input	
RUND	51	B10	Output	For debug only. No connection is required. This signal goes high for short periods < 10 ms while the decoder is processing data, and goes low when the decoder is not processing data. The signal may be helpful in diagnosing problems.
RUNE	52	E9	Output	For debug only. No connection is required. This signal goes high for short periods < 10 ms while the encoder is processing data, and goes low when the encoder is not processing data. The signal may be helpful in diagnosing problems.
CODEC_TX_CLK	53	D9	Input	Clock for I2S interface. Must connect to CODEC_RX_CLK.
CODEC_TX_FS	54	C9	Input	Frame Sync for I2S interface. Must connect to CODEC_RX_FS.
CODEC_TX	55	B9	Output	Output PCM data for I2S interface

IFRAME	64	B7	Input	50 Hz Clock Input. In codec mode, when skew control is enabled, this signal forms the 20 ms frame boundaries. A frame begins the first sample after the rising edge of IFRAME. The signal is not used in packet mode or in codec mode when skew control is disabled.
CODEC_RX_CLK	66	C7	Input	Clock for I2S interface. Must connect to CODEC_TX_CLK.
CODEC_RX_FS	67	J10	Input	Frame Sync for I2S interface. Must connect to CODEC_TX_FS.
CODEC_RX	68	C5	Input	Input PCM data for I2S interface.
UART_RTS	73	D4	Output	Flow control output signal. The signal is low when the AMBE-4020™ is ready to receive data on UART_RX. The signal is high when the AMBE-4020™ is not ready to receive data on UART_RX. Sending data to the chip when UART_RTS is high may result in errors. While RESET_n is pulled low and for a short time after it is brought high, UART_RTS becomes an input and is pulled low via a 20 k Ω resister. If it is desired to have UART_RTS high during reset then it is recommended to pull UART_RTS high using a 6.2 k Ω (or lower) resister.
UART_CTS	74	D3	Input	Flow control input signal. When the signal is low, the AMBE-4020™ is allowed to transmit data on UART_TX. When the signal is high, the AMBE-4020™ stops transmitting data on UART_TX. Note that if UART_CTS is set high while a transmission is in progress, data flow will not be stopped until transmission of the current byte completes.
UART_RX	75	C3	Input	UART receive data. This is the input data signal for a conventional UART using 8 data bits, no parity, and 1 stop bit (8N1). Hardware flow control is used. If the connected device does not support hardware flow control (consequently UART_RTS is ignored), then other means must be used to prevent overflow.
UART_TX	76	B3	Output	UART transmit data. This is the output data signal for a conventional UART using 8 data bits, no parity, and 1 stop bit (8N1). Hardware flow control is used. If hardware flow control is not supported by the connected hardware then UART_CTS should be pulled low, otherwise the chip will be prevented from transmitting packets.
VDD Digital Power	7, 38, 50, 60, 70	E5, E6, E7, L10	PWR	Digital supply voltage: This can be connected to the same supply as VDDA but should be isolated to minimize noise and error. See drawing on page 111
VSS Ground	8, 39, 49, 59, 69	F7, G3, G7, K10, L6	GND	Core and Digital I/O Pins to Ground.
No Connections (**LQFP** Chip)	1, 2, 15, 16, 23, 24, 25, 26, 27, 28, 29, 30, 31, 36, 46, 47, 48, 56, 57, 58, 61, 62, 63, 65, 71, 72, 77, 78, 79, 80			

No Connections **(BGA** Chip)	A1, A4, A5, A6, A9, A10, A11, B1, B4, B5, B6, B11, C1, C2, C6, C10, C11, D1, D2, D6, D10, D11, E1, E8, E10, E11, F3, F8, F9, F10, F11, G8, H1, H2, H3, H4, H5, H6, H8, H9, H11, J1, J2, J3, J4, J5, J6, J7, J9, K3, K4, L7			

Table 2 Pinout List

2.6 UART Interface

The serial interface supports asynchronous communication of real-time data to other asynchronous peripherals that use the standard non-return-to-zero (NRZ) format. The UART interface allows packets to be received and transmitted by the AMBE-4020™.

Pin Name	Pins		Direction	Description
	LQFP	BGA		
UART_TX	76	B3	Output	UART Transmit Data
UART_RX	75	C3	Input	UART Receive Data
UART_RTS	73	D4	Output	UART Request to Send
UART_CTS	74	D3	Input	UART Clear to Send

Table 3 UART Interface Pins

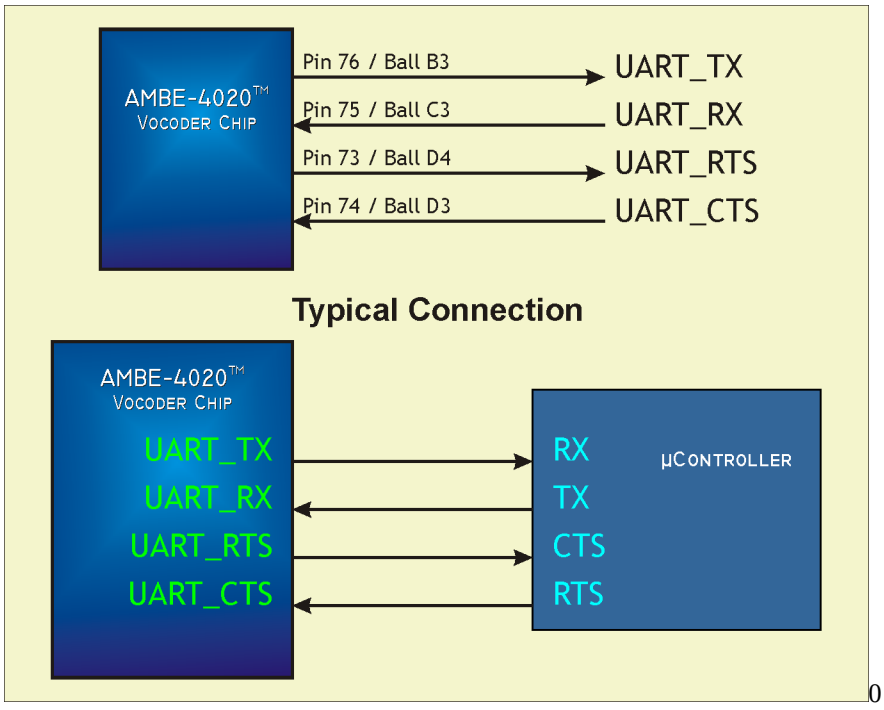


Figure 6 UART Connection

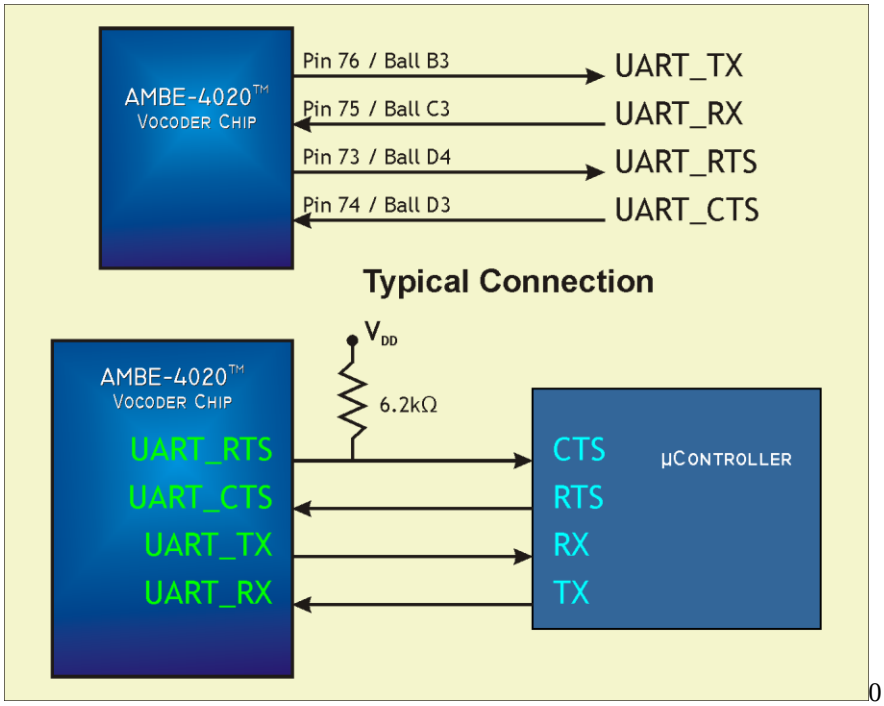


Figure 7 UART with 6.2kOhm Pull-up Resistor Connection

The AMBE-4020™ Vocoder Chip transmits packets using pin UART_TX and receives packets using pin UART _RX. Each serial word transmitted or received uses 8 data bits, no parity bits, and one stop bit.

The AMBE-4020™ supports bidirectional flow control via the UART_RTS and UART_CTS pins. When the AMBE-4020™ is ready to receive UART data, UART_RTS is low. When UART_RTS is high, no UART data should be sent to the AMBE-4020™. If UART_CTS is low, the AMBE-4020™ will transmit UART data if available. If UART_CTS is high then the AMBE-4020™ will not transmit data.

During reset, UART_RTS becomes an input signal and is pulled low via an internal 20-50 kΩ resister. Shortly after reset comes high UART_RTS is configured as an output and driven high. After the packet interface is configured and ready to receive data, UART_RTS is driven low. See Figure 22 Reset Timing. If desired, you can prevent UART_RTS from being pulled low by pulling it high using a 6.2 k Ω (or lower) pull-up resister.

The AMBE-4020™ sets UART_RTS high when the number of packets in its packet queue is ≥ $(FLOWPKT)^8$. After consuming packets, the AMBE-4020™ sets UART_RTS low when the number of packets in its packet queue is < $(FLOWPKT)^8$. The parameter $(FLOWPKT)^8$ defaults to 3 after reset for boot configuration 0-3. Boot configurations 1-3 are user programmable. In addition, in Packet Mode $(FLOWPKT)^8$ can be changed by sending a control packet containing a PKT_FLOWPKT field.

If the connected device does not support hardware flow control, then the UART_CTS pin should be pulled low, otherwise the AMBE-4020™ will not be able to transmit packets. In addition, the connected device must control the rate of packets sent to the AMBE-4020™ by other means.

The AMBE-4020™ supports baud rates between 28,800 and 750,000 baud. The most common baud rates are 28,800, 57,600, 115,200, 230,400, and 460,800 baud. Intermediate baud rates may also be specified. The baud rate, is controlled by a parameter named $(BAUD)^{24}$. Section 0 specifies the initial baud rate for each boot configuration. In addition, the PKT_BAUD control packet field can be used to change the baud rate while operating in packet mode.

Care is needed when switching the baud rate. The packet containing PKT_BAUD must be sent at the initial baud rate and the response packet will be received at the new baud rate. This means that the device that is sending the PKT_BAUD field also needs to change its baud rate after the control packet is transmitted but before its response is received. There are two ways to accomplish this.

The first method is for the device to set UART_CTS high before sending the control packet. This will prevent the AMBE-4020™ from sending the response until the device is ready. After the control packet is sent, the device should switch its own baud rate to the new baud rate, then set UART_CTS low, at which time the AMBE-4020™ will transmit the response packet at the new baud rate.

The second method to accomplish a baud rate change is for the device to transmit PKT_BAUD followed by PKT_DELAYNUS together in one control packet. The device then needs to quickly change its own baud rate and be ready to receive the response packet before the specified delay expires.

In general, baud rates below 125,000 have better power performance. 115,200 is the recommended baud rate for codec mode. In order to have adequate throughput for real-time operation in packet mode, a baud rate ≤ 250000 baud is recommended. Baud rates < approximately 172,800 baud will not have sufficient throughput for real time operation in packet mode and baud rates > 250,000 baud will consume more power.

The baud rate must be ≤ 125,000 baud prior to entering Low Power Packet Mode or Sleep Mode using a PKT_PMODE control field. The baud rate must also be ≤ 125,000 prior to entering push-to talk codec mode when $(PMODE)^8$ is set to either 1 or 2. Unexpected behavior may result when using these low power modes without adhering to these baud rate requirements.

2.7 Internal 16-bit ADC Interface

Pin Name	Pin		Direction	Description
	LQFP	BGA		
ANALOG_IN_P	13	K1	Input	Positive
ANALOG_IN_N	14	K2	Input	Negative

Table 4 16-bit ADC Interface Pins

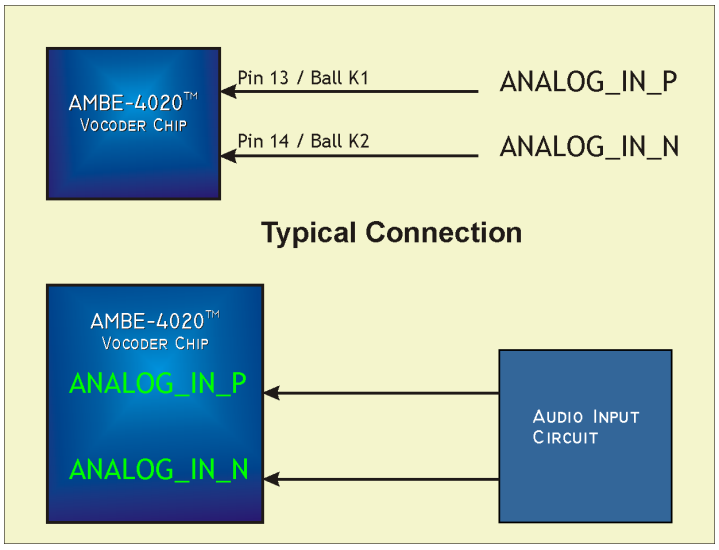


Figure 8 ADC Connection (For more details see schematic in Section 8.4)

The 16-bit ADC allows an analog signal to be received by the AMBE-4020™ through the differential analog input pins, ANALOG_IN_P and ANALOG_IN_N. The differential signal is sampled and a decimation filter is applied. In encoder codec mode, full duplex codec mode or in push-to-talk codec mode with ENC=1, these 8 kHz samples are then input to the encoder which processes the samples and produces channel packets containing compressed speech data. In encoder passthru mode, it is possible to pass the 8 kHz samples out directly in speech packets without compression.

2.8 Internal 12-bit DAC Interface

Pin Name	Pin		Direction	Description
	LQFP	BGA		
ANALOG_OUT	22	K5	Output	Analog output signal

Table 5 12-bit DAC Interface Pins

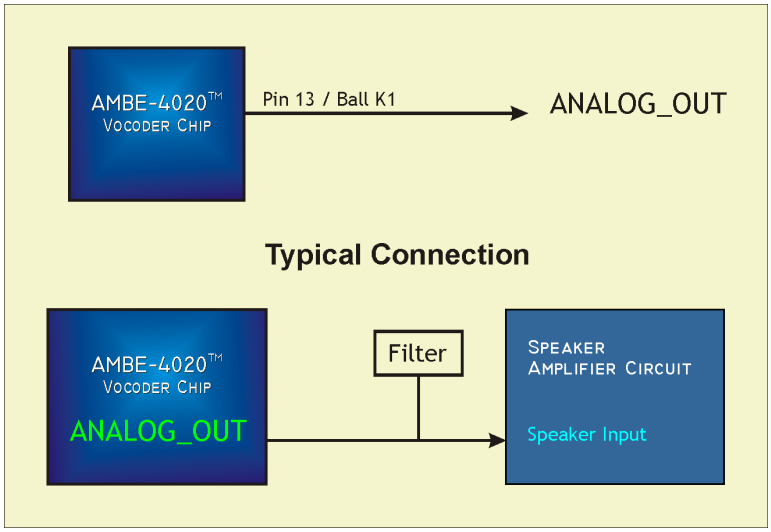


Figure 9 DAC Connection

The 12-bit DAC allows the AMBE-4020™ to output an analog signal on the pin ANALOG_OUT. In decoder codec mode, full duplex codec mode or push-to-talk codec mode with ENC=0 and DEC=1, the decoder produces an 8 kHz digital speech signal. This signal is fed into a 12-bit DAC. In decoder passthru mode, speech from speech packets are passed directly to the DAC bypassing the decoder.

2.9 Digital Mic Interface

Pin Name	Pins		Direction	Description
	LQFP	BGA		
DMIC_CLK	34	K9	Output	PDM bit clock
DMIC_RX	35	L9	Input	PDM Data from Mic

Table 6 DMIC Interface Pins

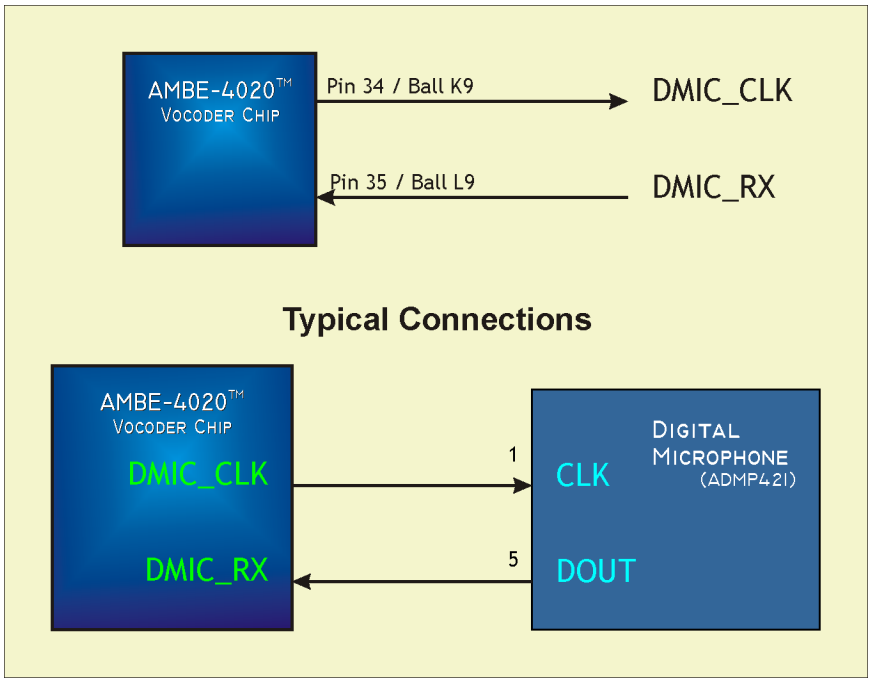


Figure 10 DMIC Connection (For more details see schematic in Section 8.4)

The digital mic interface (DMIC_CLK and DMIC_RX) receives the output of a digital microphone. The digital mic interface connection provides a robust digital signal that can help the overall product design effort by reducing external filtering components and ease the placement of the microphone with respect to the product's RF section. By using, the digital mic input of the AMBE-4020™ users can take advantage of high performance digital technologies that are less susceptible to noise than sensitive analog signal processing.

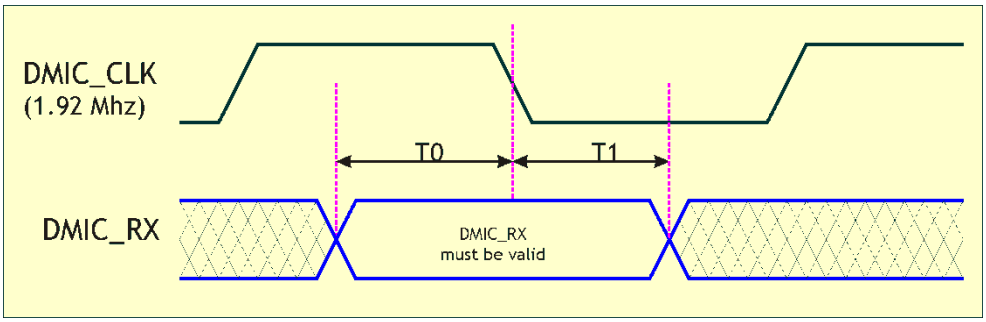


Figure 11 DMIC Timing

ID	Characteristic	Minimum	Maximum	Unit
T0	DMIC_RX set-up before CODEC_CLK	30		ns
T1	DMIC_RX hold after CODEC_CLK	6.5		ns

Table 7 DMIC Timing

A 1.92 MHz clock signal is output on DMIC_CLK. The digital microphone produces PDM samples at 1.92 MHz that are received on the DMIC_RX pin. The AMBE-4020™ reads DMIC_RX at each falling edge of DMIC_CLK. The 1-bit at 1.92 MHz data is filtered and down-sampled to obtain 16-bit linear PCM. Then another down-sampling filter is applied in order to

obtain an 8 KHz signal that is input to the encoder. In encoder codec mode, full duplex codec mode or push-to-talk codec mode with ENC=1, this 8 kHz signal is encoded and the resulting compressed speech data is output in the form of channel packets. In encoder passthru mode, the filtered 8 kHz speech data from the DMIC can be passed out directly via speech packets bypassing the encoder.

2.10 ENC/DEC Push-to-Talk Duplex Control Pins

The ENC and DEC pins determine the duplex of the vocoder when the AMBE-4020™ is configured for push-to-talk codec mode. Note that push-to-talk codec mode is always half-duplex - at any given time the encoder or decoder runs, but never both at the same time.

When ENC is high, the encoder runs. When DEC is high and ENC is low, the decoder runs. When ENC and DEC are both low, neither the encoder nor the decoder are active; three choices are available for power savings during periods of inactivity:

Choice 1
Sleep Mode, when $(PMODE)^8 = 2$, offers the greatest power savings, but longest wakeup time.

Choice 2
Low Power Mode, when $(PMODE)^8 = 1$, provides moderate power savings and moderate wakeup time.

Choice 3
Idle Mode, when $(PMODE)^8 = 0$, provides the lowest power savings, but lowest wakeup time.

When the encoder is active, the AMBE-4020™ outputs a channel packet once every 20 ms.

When the decoder is active, the AMBE-4020™ consumes a channel packet once every 20 ms. If no channel packets are received, the decoder produces comfort noise samples.

Channel packets received while the encoder/decoder are inactive are discarded.

2.10.1 Starting the Encoder or Decoder via the ENC/DEC pins

When neither the encoder or decoder are running (ENC = DEC = 0), the AMBE-4020™ remains idle in a low power state as determined by the setting for $(PMODE)^8$. A transition of either the ENC or the DEC pin from low to high will initiate wakeup from the low power state and subsequent start up of either the encoder or decoder.

For $(PMODE)^8 = 2$, wakeup takes less than 4.5 ms.

For $(PMODE)^8 = 1$, wakeup takes less than 2.5 ms.

For $(PMODE)^8 = 0$, wakeup takes less than 0.5 ms.

After wakeup, when skew control is disabled, OFRAME is set high immediately and will cycle with a 20 ms period 50% duty cycle thereafter. When the encoder is started, the first channel packet is transmitted after the falling edge of OFRAME in the second frame period, i.e. approximately 30 ms after the initial rising edge of OFRAME. When the decoder is started, it will consume a channel packet after the first falling edge of OFRAME, if the entire packet was received before the falling edge of OFRAME.

After wakeup, when skew control is enabled, OFRAME is set high upon the first rising edge of IFRAME. Thus, the startup time for the encoder or decoder could be delayed by up to 20 ms due to the need to synchronize with IFRAME. This variable sync time could be avoided by setting the ENC/DEC pin synchronously with IFRAME.

2.10.2 Stopping the Encoder or Decoder via the ENC/DEC pins

When either the encoder or the decoder is running, the ENC/DEC pins are sampled at the falling edge of OFRAME to determine when the vocoder function (encoder or decoder) should be shut off.

If the encoder is running and ENC=0 is seen at the falling edge of OFRAME, then the encoder shutdown is initiated. The AMBE-4020™ will require an additional OFRAME cycle to shut down the encoder and flush out channel packets in the buffer.

If the decoder is running and DEC=0 is seen at the falling edge of OFRAME, then the decoder shutdown is initiated. The AMBE-4020™ may require up to an additional $(NCHANPKT)^8$ +3 OFRAME cycles before the decoder is shut down. Note that after the encoder or decoder is shut down, the OFRAME signal will remain in the state OFRAME = 0.

2.10.3 Using ENC pin to switch between encoder and decoder with DEC pin held high

When it is not desired to enter a low power state during periods when the encoder is not running, the DEC pin may be held high. If this is the case then, when the ENC pin is held high the encoder runs, and when ENC is brought low the decoder runs. In this case, it is acceptable to run the decoder even when no channel packets are being sent to the AMBE-4020™. In the absence of received channel packets, the AMBE-4020™ decoder will produce low level comfort noise. Note that when this method is used, there will be a discontinuity in the normal periodic OFRAME timing as the vocoder function is changed from encoder to decoder or vise versa.

2.10.4 Treatment for Startup Overlapped with Shutdown

Sections 2.10.1 and 2.10.2 describe how the encoder and decoder can be started or stopped by changing the ENC/DEC pins. If an attempt is made to start the encoder or decoder while a shutdown of the encoder or decoder is underway, the startup will be delayed until the shutdown has completed.

To avoid overlapped shutdown/startup allow the encoder or decoder to finish shutting down before raising either the ENC or DEC pin. After the falling edge of ENC, waiting 60 ms before setting either ENC or DEC high will guarantee that the encoder has completed its shutdown before starting the decoder (or restarting the encoder). After the falling edge of DEC, waiting $80 + 20*(NCHANPKT)^8$ ms (default is 100 ms) before setting either ENC or DEC high will guarantee that the decoder has completed its shutdown before starting the encoder (or restarting the decoder).

2.11 I2S Interface

The I2S interface allows the AMBE-4020™ to receive and transmit 8 kHz speech data. Typically, the I2S signal is connected to an external codec. It could also be connected to any other device that is capable of sending and/or receiving serial speech samples. The clocks and frame syncs are generated by the codec or other device. The sampling rate must be 8 kHz.

Pin Name	Pins		Direction	Description
	LQFP	BGA		
CODEC_TX	55	B9	Output	Output PCM data
CODEC_RX	68	C5	Input	Input PCM data
CODEC_TX_CLK	53	D9	Input	I2S Interface Clock
CODEC_RX_CLK	66	C7	Input	I2S Interface Clock
CODEC_TX_FS	54	C9	Input	I2S Interface frame sync

CODEC_RX_RS	67	J10	Input	I2S Interface frame sync

Table 8 I2S Interface Pins

When the I2S interface is selected, the CODEC_TX_CLK, CODEC_RX_CLOCK, CODEC_TX_FS, and CODEC_RX_FS signals should be active prior to entry into codec mode. If proper clocks and framing signals are not available a watchdog reset may result upon entry into codec mode.

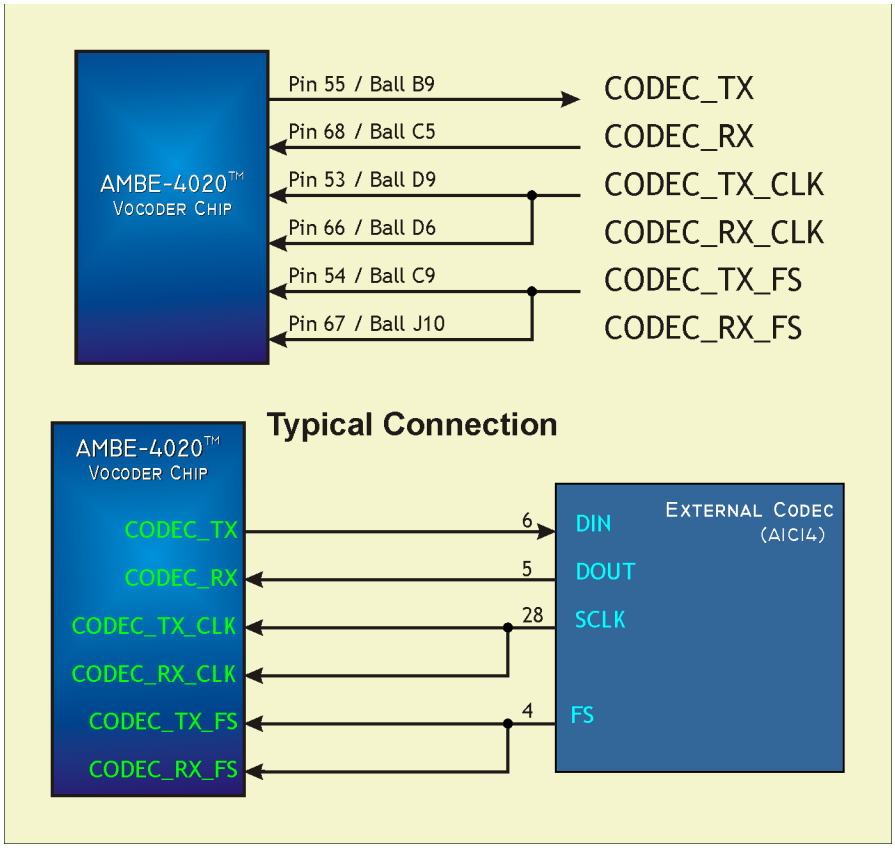


Figure 12 I2S Connection

The I2S interface can transfer 16-bit PCM data, 8-bit μ-law, or 16-bit A-law. This selection is controlled by a parameter named (COMPAND)[8] which can be selected using the PKT_COMPAND packet field or by pre-programming the selection into the boot configuration.

Typically, the clock rate used for 16-bit PCM data is 128 kHz (16 × 8 kHz), however clock rates as high as 3.072 MHz are possible, as long as the sample rate remains 8 kHz.

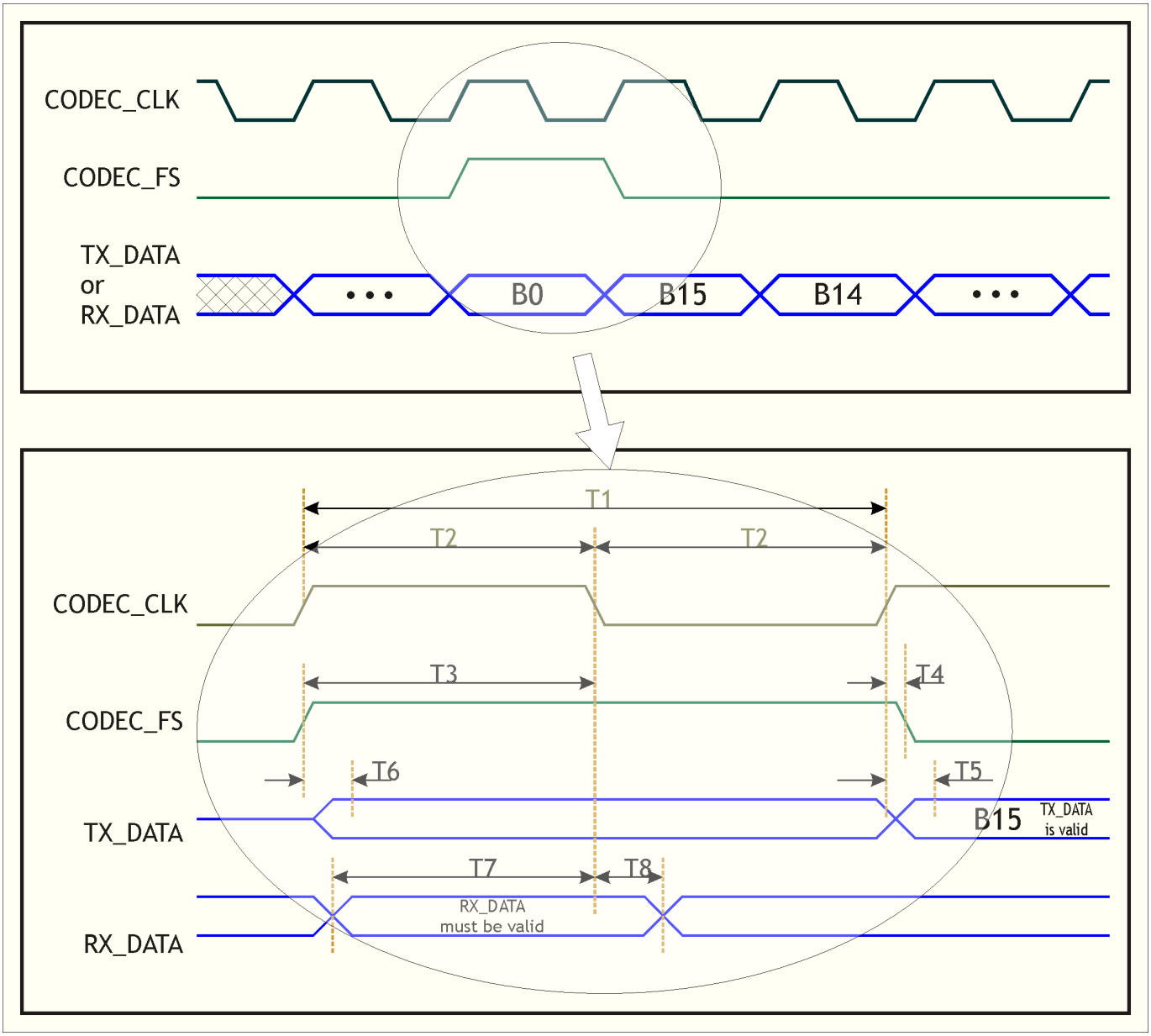


Figure 13 I2S Timing diagram

ID	Characteristic	Minimum	Maximum	Unit
T1	CODEC_CLK Cycle Time	325	15625	ns
T2	CODEC_CLK pulse width High / Low	45%	55%	CODEC_CLK period
T3	CODEC_FS Set-up before CODEC_CLK	30		ns
T4	CODEC_FS input hold after CODEC_CLK	7.6		ns
T5	CODEC_CLK to TX_DATA output valid		67	ns
T6	CODEC_FS input assertion to TX_DATA output valid		72	ns
T7	RX_DATA set-up before CODEC_CLK	30		ns
T8	RX_DATA hold after CODEC_CLK	6.5		ns

Table 9 I2S Timing

In encoder codec mode, full duplex codec mode or in push-to-talk codec mode with ENC=1, the encoder receives 8 kHz speech data from CODEC_RX, encodes it and produces channel packets containing compressed speech data. In encoder passthru mode, the speech data can be passed out directly via speech packets.

In decoder codec mode, full duplex codec mode or in push-to-talk codec mode with ENC=0 and DEC=1, the decoder receives channel packets containing compressed speech data. The decoder then produces 8 kHz speech data that is clocked out on CODEC_TX.
In decoder passthru mode, 8 kHz speech data is received via speech packets and then clocked out on CODEC_TX.

2.12 I2C Interface

The AMBE-4020™ operates as an I2C master device and may be used to configure an I2C slave device. The I2C interface is primarily intended as a means of configuring a Texas Instruments AIC14 codec or similar device, but may also be able to configure other I2C slave devices. All of the default settings in boot configuration 0 are setup for configuration of the AIC14.

Pin Name	Pins		Direction	Description
	LQFP	BGA		
I2C_SDA	11	G1	I/O	Configuration data to an external codec
I2C_SCL	12	G2	I/O	Configuration data to an external codec

Table 10 I2C Interface Pins

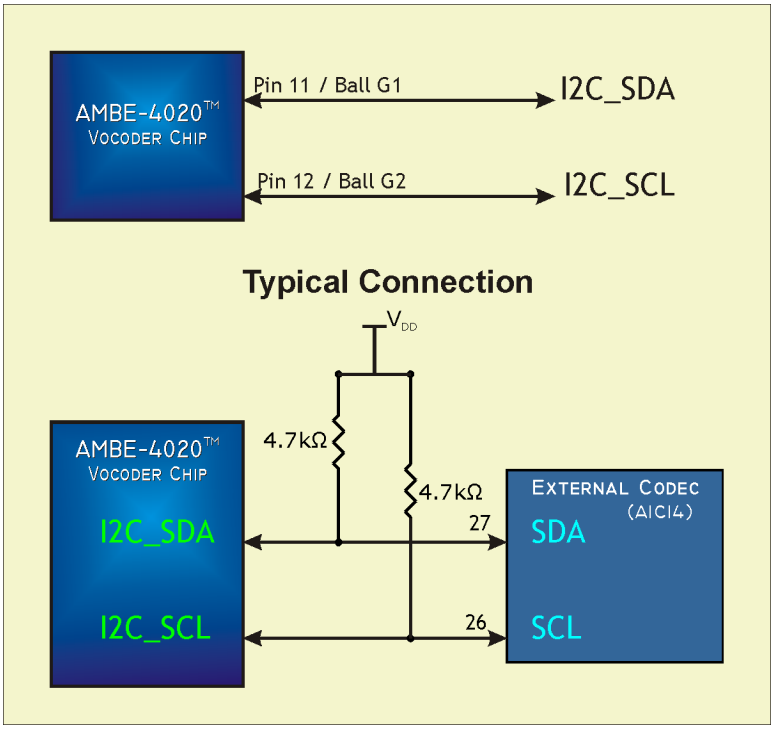


Figure 14 I2C Connection

The default SCL clock rate is 50 KHz.

The AMBE-4020™ supports two different methods for configuring I2C slave devices:

Method 1:
A UART packet field named PKT_I2CDATA is used to specify the number of registers to be written on the slave device and specifies register/value pairs for each register to be written. If no PKT_I2CDATA field is received then the default register/value pairs will be sent. In addition, UART packet field PKT_CONFIGI2C may be used to specify 3 parameters: (I2CADDR)[8], (I2CDELAY)[8], and (I2CDIV)[8].

Send PKT_I2CDATA to configure the number of I2C registers to be written and to specify the register/value pairs.
Send PKT_CONFIGI2C to configure the slave address, reset delay, and SCL clock rate.
Send PKT_STARTCODEC and specify duplex=encoder with I2S as input or specify duplex=decoder with I2S output.

Upon entry into codec mode, if the I2S interface is the selected, the AMBE-4020™ sets the CODEC_RESETn signal high, then waits for a programmable delay for the device to be ready to receive I2C configuration data. After this delay, the AMBE-4020™ uses the I2C protocol to write to 0-16 registers with the pre-configured data.

Method 2:
A UART packet field named PKT_WRITEI2C is used to configure registers. Each PKT_WRITEI2C field maybe used to write to 1 or more consecutive registers (if the device supports writing to multiple registers). The slave address is a parameter of the PKT_WRITEI2C field. This method makes it easy to write to multiple slave devices with differing addresses. Since PKT_WRITEI2C may write to multiple registers and PKT_WRITEI2C may be repeated any number of times, it can be used to write any number of registers, whereas method 1 is limited to a maximum of 16 registers. In addition, a packet field named PKT_READI2C may be used to read 1 or more registers from a specified slave address.

2.13 Boot Configuration Pins

(BOOT0 and BOOT1 LQFP pins 32 and 33, BGA pins L8 and K8)

The boot configuration pins select one of four boot configurations. Each boot configuration specifies the initial state of various parameters such as the baud rate, vocoder rate, codec mode vs. packet mode, and formatting for output packets. Boot configuration 0 is fixed and may not be changed. Boot configurations 1, 2, and 3 are programmed with factory defaults stored in persistent memory. These configurations may be read or written using packet field PKT_BOOTCFG. Table 12: Factory Settings for each Boot Configuration, shows the factory settings of each boot configuration. To guarantee data retention of more than 20 years, each boot configuration must be written less than 30,000 times.

The BOOT0 and BOOT1 pins have internal pull-down resisters. If it is desired to use only boot configuration 0, then it is possible to leave the BOOT0/BOOT1 bins unconnected.

Parameter	Parameter Description	Brief Summary
(BAUD)[24]	PKT_BAUD	Specify UART baud rate
(SPCHFMT)[32]	PKT_SPCHFMT	Specifies format for outgoing speech packets
(CHANFMT)[32]	PKT_CHANFMT	Specifies format for outgoing channel packets
(ECONTROL)[16]	PKT_ECONTROL	Specifies encoder control flags
(DCONTROL)[16]	PKT_DCONTROL	Specifies decoder control flags
(DISCARD)[16]	PKT_DISCARD	Specifies the number of initial samples to discard
(GDIV)[16]	PKT_GDIV	Specifies divider used to generate GFRAME
(COMPAND)[8]	PKT_COMPAND	Specifies whether companding is used and choose A-law or μ-law
(RATET)[8]	PKT_RATET	Specifies vocoder rate
(CODECCFG)[8]	PKT_STARTCODEC	Selects codec mode duplex, interfaces, passthru, skew
(FLOWPKT)[8]	PKT_FLOWPKT	Specifies the max number of packets before UART_RTS is set.
(NCHANPKT)[8]	PKT_NCHANPKT	Specifies the number of channel packets to buffer in codec mode.
(IGAIN)[8]	PKT_GAIN	Specifies gain applied during encoder

(OGAIN)[8]	PKT_GAIN	Specifies gain applied during decoder
(UFRAME_HI)[8]	PKT_UFRAME	Specifies when UFRAME rises relative to rising edge of OFRAME.
(UFRAME_LO)[8]	PKT_UFRAME	Specifies when UFRAME falls relative to rising edge of OFRAME.
(BREAKF)[8]	PKT_BREAKF	Specifies what happens when a UART break signal is received
(MODE)[8]	PKT_STARTCODEC, PKT_STOPCODEC	Specifies Packet Mode or Codec Mode
(PARITYMODE)[8]	PKT_PARITYMODE	Specifies whether the chip outputs parity fields in outgoing packets and checks for parity fields in incoming packets.
(PMODE)[8]	PKT_PMODE	Specifies power mode
$(RCW_0)^{16}$ - $(RCW_5)^{16}$	PKT_RATEP	Specifies custom rate words used when $(RATET)^8 = 64$
(I2CADDR)[8]	PKT_CONFIGI2C	Specifies I2C slave address
(I2CDELAY)[8]	PKT_CONFIGI2C	Specifies delay after codec is reset, before I2C registers are written
(I2CDIV)[8]	PKT_CONFIGI2C	Dividers control for configuration of I2C clock rate
(I2CNREG)[8]	PKT_I2CDATA	Specifies the number of registers to be written to via I2C
$(I2CDATA_0)^8$ - $(I2CDATA_{31})^8$	PKT_I2CDATA	Specifies register number/value pairs for each register to be written.
(ECHOSUPLIM)[8]	PKT_ECHOSUPLIM	Specifies the maximum attenuation in dB applied by the echo suppressor. . $0 \leq (ECHOSUPLIM)^8 \leq 60$.
(ECHOLEN)[16]	PKT_ECHOLEN	Specifies the length of the echo canceller in samples. $0 \leq (ECHOLEN)^{16} \leq 128$. $(ECHOLEN)^{16}$ must be evenly divisible by 8. 8 samples is equivalent to 1 ms, such that 128 samples equates to a 16 ms echo canceller.

Table 11: Parameters Specified for each Boot Configuration

Parameter	Fixed Configuration	User Programmable Configurations		
	Boot Configuration 0 BOOT1=0,BOOT0=0 (Fixed)	Boot Configuration 1 BOOT1=0,BOOT0=1 (R/W)	Boot Configuration 2 BOOT1=1,BOOT0=0 (R/W)	Boot Configuration 3 BOOT1=1,BOOT0=1 (R/W)
(BAUD)[24]	0x01C200	0x038400	0x01C200	0x01C200
(SPCHFMT)[32]	0x00000000	0x00000000	0x00000000	0x00000000
(CHANFMT)[32]	0x00000000	0x00000000	0x00015004	0x00015004
(ECONTROL)[16]	0x0040	0x0040	0x0040	0x0040
(DCONTROL)[16]	0x0000	0x0000	0x0000	0x0000
(DISCARD)[16]	0x0000	0x0000	0x0000	0x0000
(GDIV)[16]	0x0000	0x0271	0x0271	0x0271
(COMPAND)[8]	0x00	0x00	0x00	0x00
(RATET)[8]	0x00	0x00	0x00	0x00
(CODECCFG)[8]	0x00	0x14	0x14	0x14
(FLOWPKT)[8]	0x03	0x03	0x03	0x03
(NCHANPKT)[8]	0x01	0x01	0x02	0x02
(IGAIN)[8]	0x00	0x00	0x00	0x00
(OGAIN)[8]	0x00	0x00	0x00	0x00
(UFRAME_HI)[8]	0x28	0x28	0x64	0x28
(UFRAME_LO)[8]	0x78	0x78	0x19	0x78

(BREAKF)[8]	0x01	0x01	0x01	0x01
(MODE)[8]	0x00	0x00	0x01	0x01
(PARITYMODE)[8]	0x01	0x01	0x00	0x01
(PMODE)[8]	0x00	0x00	0x02	0x00
(RCW_0)[16] - (RCW_5)[16]	0x0000 0x0000 0x0000 0x0000 0x0000 0x0000	0x0000 0x0000 0x0000 0x0000 0x0000 0x0000	0x0000 0x0000 0x0000 0x0000 0x0000 0x0000	0x0000 0x0000 0x0000 0x0000 0x0000 0x0000
(I2CADDR)[8]	0x80	0x80	0x80	0x80
(I2CDELAY)[8]	0x01	0x01	0x01	0x01
(I2CDIV)[8]	0x14	0x14	0x14	0x14
(I2CNREG)[8]	0x07	0x07	0x07	0x07
($I2CDATA_0$)[8] - ($I2CDATA_{31}$)[8]	0x02 0x20 0x01 0x61 0x03 0x05 0x04 0x81 0x04 0x63 0x05 0xBB 0x06 0x04 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF	0x02 0x20 0x01 0x61 0x03 0x05 0x04 0x81 0x04 0x63 0x05 0xBB 0x06 0x04 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF	0x02 0x20 0x01 0x61 0x03 0x05 0x04 0x81 0x04 0x63 0x05 0xBB 0x06 0x04 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF	0x02 0x20 0x01 0x61 0x03 0x05 0x04 0x81 0x04 0x63 0x05 0xBB 0x06 0x04 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF
(Reserved)[8]	0x01 or 0x00	0x01 or 0x00	0x01 or 0x00	0x01 or 0x00
(ECHOSUPLIM)[8]	0x1E or 0x00	0x1E or 0x00	0x1E or 0x00	0x1E or 0x00
(ECHOLEN)[16]	0x80 or 0x00	0x80 or 0x00	0x80 or 0x00	0x80 or 0x00

Table 12: Factory Settings for each Boot Configuration

2.14 GPIO Pins

Pin Name	Pins		Direction	Description
	LQFP	BGA		
GPIO0	4	F4	I/O	general purpose input/output pin
GPIO1	5	H7	I/O	general purpose input/output pin
GPIO2	6	G4	I/O	general purpose input/output pin
GPIO3	9	F1	I/O	general purpose input/output pin

Table 13: GPIO Interface Pins

By default, all GPIOs are disabled after reset. Each of the four GPIO pins may be enabled/disabled independently and the direction of each pin can be specified independently. All control over the GPIO pins is requested by sending the packet field PKT_GPIO. The values of output pins may be specified and the values of input pins may be read. It is also possible to send a

packet which waits for the GPIO pin(s) to be in a specified state, at which time a response packet is sent. Refer to PKT_GPIO for details.

2.15 Crystal / Oscillator Usage

The AMBE-4020™ Vocoder Chip has an on-chip, PLL-based clock module and requires an input clock frequency of 4 MHz. The PLL-based clock module provides all the necessary clocking signals for the device, as well as control for low-power mode entry. The AMBE-4020™ Vocoder Chip two modes of operation:

To use a crystal oscillator with the AMBE-4020™ Vocoder Chip, connect the crystal across EXTAL and along with one external capacitor from each of these pins to ground.

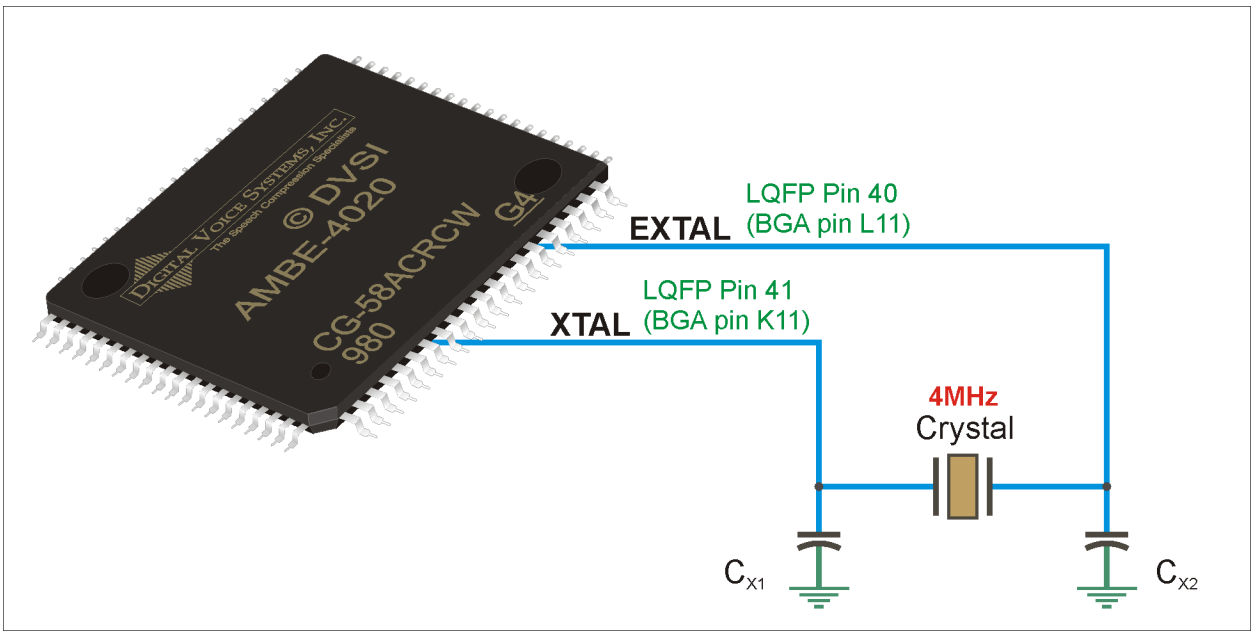


Figure 15 Internal Oscillator Crystal/Ceramic Resonator

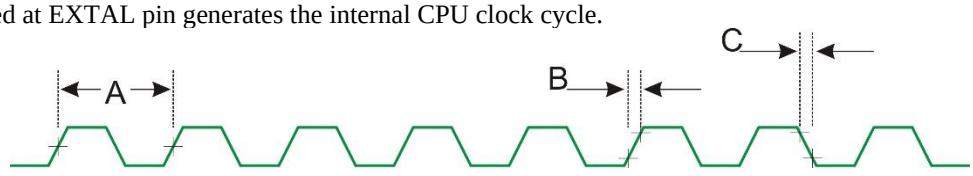
Note: the values for C_{X1} and C_{X2} are determined by the crystal manufacturer.

The following points should be noted when designing any printed circuit board layout:

- Keep EXTAL and XTAL away from high frequency digital traces to avoid coupling.
- Keep the crystal and external capacitors as close to the EXTAL and XTAL pins as possible to minimize board stray capacitance.

2.15.1 Input Clock Requirements

The clock provided at EXTAL pin generates the internal CPU clock cycle.



ID	Parameter	Typical	Unit
A	$t_{c(Cl)}$ Cycle time, EXTAL	250	ns
B	$t_{r(Cl)}$ Rise time, EXTAL	6	ns
C	$t_{f(Cl)}$ Fall time, EXTAL	6	ns

		Min	Max	Unit
	$t_{w(ClL)}$ Pulse duration EXTAL **Low** as a percentage of $t_{c(Cl)}$	40	60	%
	$t_{w(ClH)}$ Pulse duration EXTAL **High** as a percentage of $t_{c(Cl)}$	40	60	%

Parameter		Min	Nom	Max	Unit
V_{IH}	High-level input voltage EXTAL (@50uA max)	0.7 ×VDD	-	VDD	V
V_{IL}	Low-level input voltage EXTAL (@50uA max)			0.3 ×VDD	V
	Recommended frequency stability	±50			ppm

Table 14 Input Clock Requirements

2.15.2 External Clock Source

When an external source is used as the clock input, connect XTAL and EXTAL as follows:

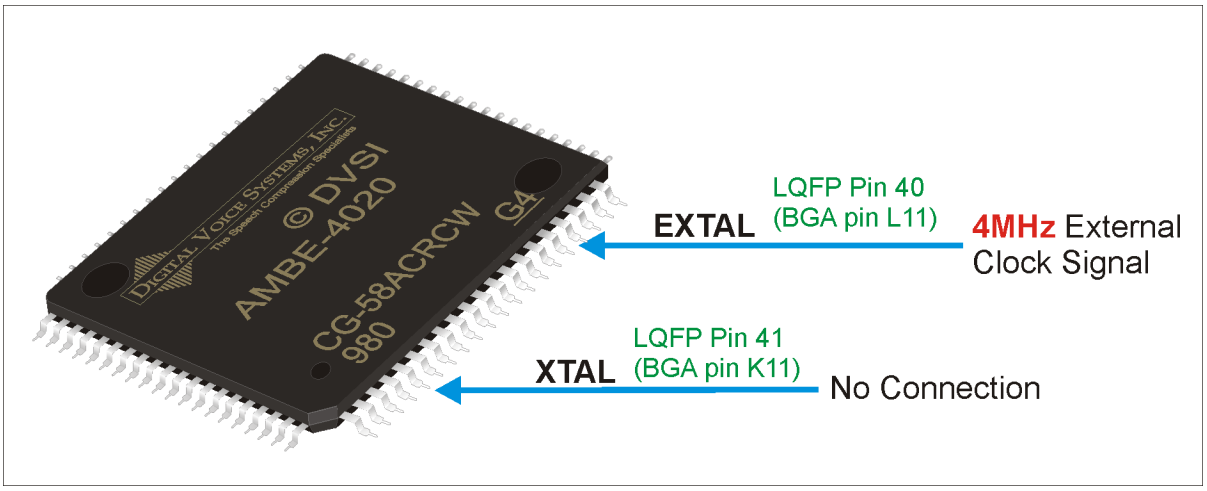


Figure 16 External Clock Connection

2.16 OFRAME/IFRAME/UFRAME/GFRAME Framing Signals

Pin Name	Pins		Direction	Description
	LQFP	BGA		
OFRAME	37	H10	Output	output framing signal
IFRAME	64	B7	Input	input framing signal
UFRAME	10	F2	Output	user customizable output framing signal
GFRAME	43	G11	Output	generated framing signal

Table 15: OFRAME/IFRAME/UFRAME/GFRAME Interface Pins

Each of these framing signals is a 50 Hz (nominal) clock. The encoder inputs or the decoder outputs 8 kHz speech samples in 20 ms (nominal) frames. The number of samples per frame is 160 (nominal). In codec mode, the AMBE-4020™ maintains a sample counter that keeps track of the number of samples per frame, such that the counter increments at a rate of 8 kHz. The clock source for the counter is dependent upon which speech interface is selected. The counter is reset to 0 after the last sample in the frame is input or output.

In codec mode (or push-to-talk codec mode), the OFRAME signal rises when the sample counter is 0 and falls when the counter is 80. When the decoder is running, it consumes a packet (when available) when the counter reaches 80. When the encoder is running, it begins outputting a channel packet when the counter is 81. If skew control is disabled, then the counter automatically resets to 0, after it reaches 160. If skew control is enabled, then the counter resets upon the rising edge of the IFRAME signal.

In codec mode (or push-to-talk codec mode) with skew control enabled, IFRAME is presumed to be a 50 Hz framing signal. Two successive rising edges of IFRAME are used to delimit a frame. IFRAME must be constrained such that the number of samples between any two rising edges is always between 156 and 164 (inclusive). The counter is reset to 0 upon the rising edge of IFRAME. The falling edge of IFRAME is not critical.

Upon entry into codec mode, the first frame interval does not begin until the first IFRAME rising edge is encountered, at which time the counter is set to 0 and begins counting. On subsequent rising edges of IFRAME, the value of the counter is saved prior to resetting the counter. The saved counter is used by the encoder/decoder in order to adjust the number of samples consumed by the encoder or produced by the decoder. IFRAME should be a stable 50 Hz signal such that frames normally contain 160 samples with an occasional frame having either 159 or 161 samples to compensate for clock skew between IFRAME and the 8 kHz sample clock.

Note that when skew control is enabled, the OFRAME signal is still output: It rises when the counter is set to 0 and falls when the counter reaches 80. The decoder consumes a frame (if available) when the counter reaches 80. The encoder begins outputting a channel packet when the counter reaches 81. The rising edge of OFRAME tracks the rising edge of IFRAME such that the rising edge of OFRAME occurs within 125 us of the rising edge of IFRAME. The falling edge of IFRAME is not tracked by OFRAME.

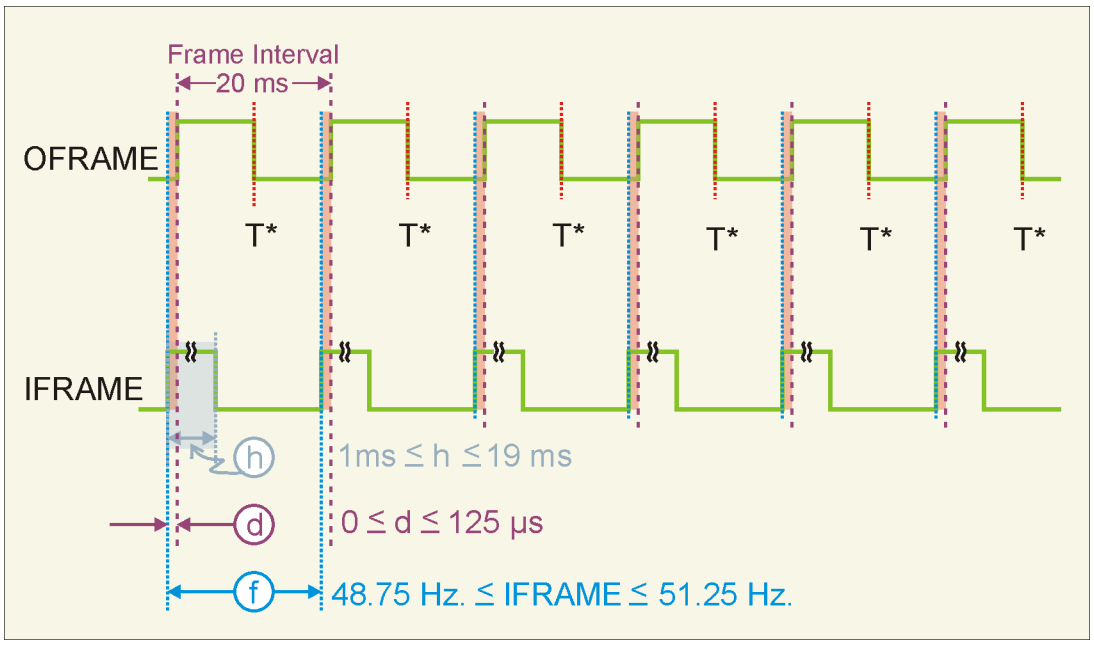


Figure 17 OFRAME Frame Interval

UFRAME is a user configurable framing signal. UFRAME is synchronous with OFRAME. Two user configurable parameters, $(UFRAME\_HI)^8$ and $(UFRAME\_LO)^8$ specify the counter value when UFRAME rises and falls. By default for boot configuration 0, UFRAME is specified to rise when the counter reaches 40 and to fall when the counter reaches 120, such that UFRAME is simply OFRAME delayed by 5 ms. UFRAME may be useful to synchronize other events running on another processor with the frame rate of the AMBE-4020™. $(UFRAME\_HI)^8$ and $(UFRAME\_LO)^8$ may be specified using the packet field PKT_UFRAME or by setting up a custom boot configuration.

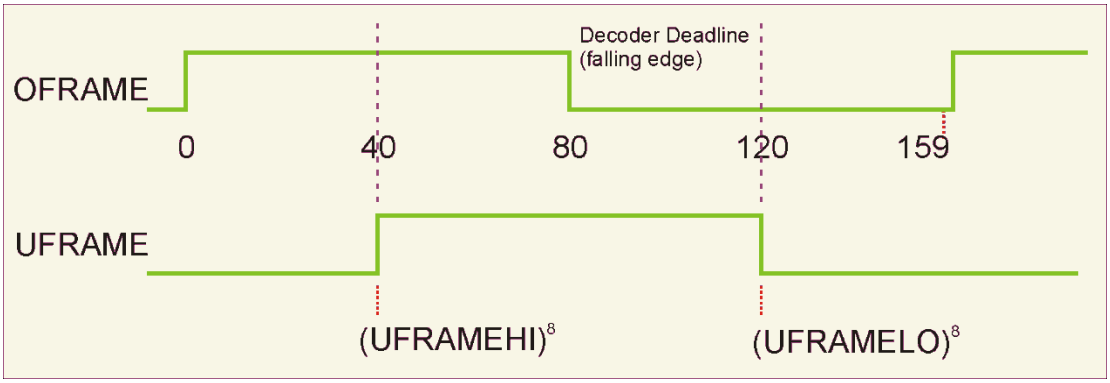


Figure 18 UFRAME

Figure 18 UFRAME shows one possible configuration where $(UFRAMEHI)^8 = 40$ and $(UFRAMELO)^8 = 120$.

GFRAME is another user configurable signal. The AMBE-4020™ outputs a framing signal on GFRAME where the frequency of GFRAME is 31250 / $(GDIV)^8$ Hz. When $(GDIV)^8 = 0$, GFRAME is disabled. When $(GDIV)^8$ is 625, the frequency of GFRAME is 50 Hz. GFRAME operates independently from the sample counter mentioned above. It effectively operates using a 31250 Hz internal clock to clock second counter that resets to 0 upon reaching $(GDIV)^8$, rises when the counter reaches 0, and falls when the counter reaches reaching $(GDIV)^8/2$. GFRAME could be used as the source of IFRAME. In packet mode, GFRAME can be used by another processor to operate the encoder/decoder at an appropriate frame rate.

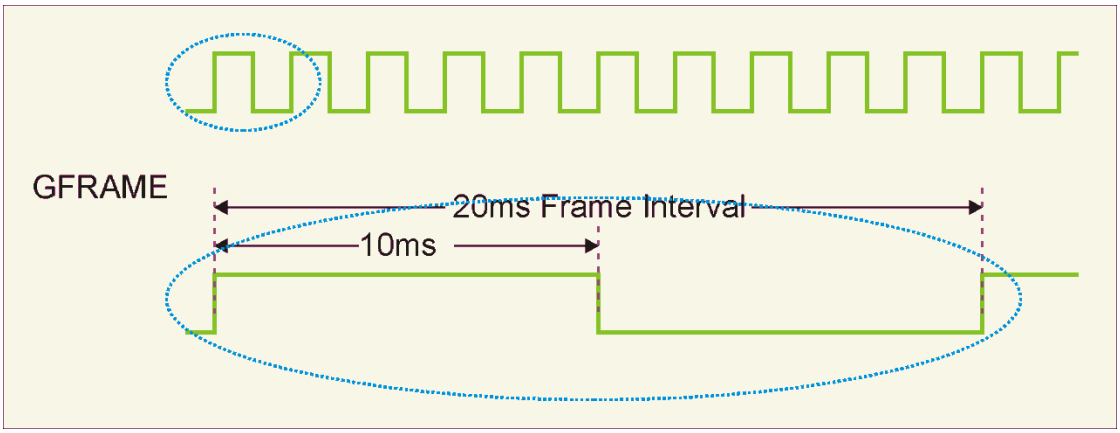


Figure 19 GFRAME

In packet mode, the functions of OFRAME/IFRAME/UFRAME/GFRAME are as follows. OFRAME is set high each time the encoder/decoder starts and set low about halfway through. There is no guaranteed period since the timing of the encoder/decoder are dependent upon when packets are received. IFRAME is not used in packet mode and has no effect. UFRAME will always be low. GFRAME operates as an independent mechanism for generating a 50 Hz frame signal.

OFRAME/UFRAME/GFRAME are all disabled during halt/sleep modes.

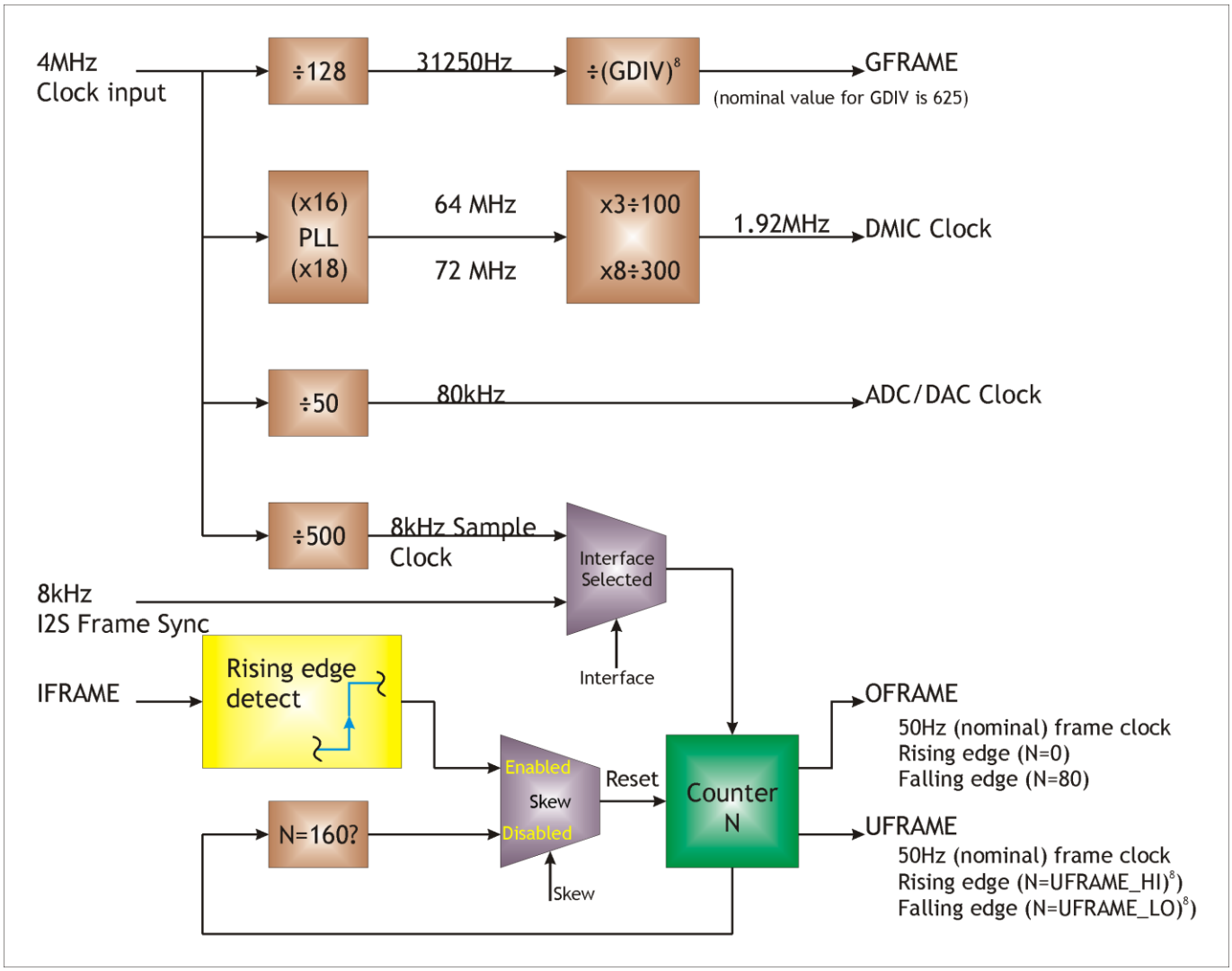


Figure 20 AMBE-4020™ Clocking

SECTION 3

3 Normal Operating Conditions

3.1 Moisture handling ratings

The length of time the AMBE-4020™ can be safely exposed to the ambient environment prior to high temperature reflow soldering follows the JEDEC industry standard classification for Moisture Sensitivity Level.

Symbol	Description	Min.	Max.	Notes
MSL	Moisture sensitivity level	--	3	1

Table 16 Moisture Sensitivity Rating

1. Determined according to IPC/JEDEC Standard J-STD-020, Moisture/Reflow Sensitivity Classification for Non-hermetic Solid State Surface Mount Devices.

3.2 ESD handling ratings

To avoid damage from the accumulation of a static charge, industry standard electrostatic discharge precautions and procedures must be employed during handling and mounting.

Symbol	Description	Min.	Max.	Notes
V_{HBM}	Electrostatic discharge voltage, human body model	-2000V	+2000V	[1]
V_{CDM}	Electrostatic discharge voltage, charged-device model	-500V	+500V	[2]
I_{LAT}	Latch-up current at ambient temperature of 105°C	-100mA	+100mA	

[1] Determined according to JEDEC Standard JESD22-A114, Electrostatic Discharge (ESD) Sensitivity Testing Human Body Model (HBM).
[2] Determined according to JEDEC Standard JESD22-C101, Field-Induced Charged-Device Model Test Method for Electrostatic-Discharge-Withstand Thresholds of Microelectronic Components.

Table 17 Electrostatic Discharge Ratings

3.3 Pin Capacitance Attributes

Symbol	Description	Min	Max	Unit
C_{IN-A}	Input capacitance: analog pins		7	pF
C_{IN-D}	Input capacitance: digital pins		7	pF

Table 18 Pin Capacitance Attributes

3.4 Thermal handling ratings

Symbol	Description	Min.	Max.	Unit	Notes
T_{STG}	Storage temperature	-55	150	°C	1
T_{SDR}	Solder temperature, lead-free	--	260	°C	2

1. Determined according to JEDEC Standard JESD22-A103, High Temperature Storage Life.
2. Determined according to IPC/JEDEC Standard J-STD-020, Moisture/Reflow Sensitivity Classification for Non-hermetic Solid State Surface Mount Devices.

Table 19 Thermal Ratings

3.4.1 Thermal Operating Requirements

Symbol	Description	Min.	Max.	Unit	Notes
T_J	Die Junction Temperature Range	-40	125	°C	
T_A	Ambient Temperature	-40°	105	°C	

Table 20 Thermal Operating Requirements

3.4.2 Thermal Attributes

Thermal Resistance Characteristics			Package Type		Unit	Note
Board Type	Symbol	Description	LQFP	BGA		
Single-layer (1s)	RθJA	Thermal resistance, junction to ambient (natural convection)	51	74	°C/W	1,2
Four-layer (2s2p)	RθJA	Thermal resistance, junction to ambient (natural convection)	36	42	°C/W	1,3
Single-layer (1s)	RθJMA	Thermal resistance, junction to ambient (200 ft./ min. air speed)	41	62	°C/W	1,3
Four-layer (2s2p)	RθJMA	Thermal resistance, junction to ambient (200 ft./min. air speed)	30	38	°C/W	1,3
--	RθJB	Thermal resistance, junction to board	20	23	°C/W	4
--	RθJC	Thermal resistance, junction to case	10	19	°C/W	5
--	ΨJT	Thermal characterization parameter, junction to package top outside center (natural convection)	2	4	°C/W	6

1. Junction temperature is a function of die size, on-chip power dissipation, package thermal resistance, mounting site (board) temperature, ambient temperature, airflow, power dissipation of other components on the board, and board thermal resistance.
2. Determined according to JEDEC Standard JESD51-2, Integrated Circuits Thermal Test Method Environmental Conditions—Natural Convection (Still Air) with the single layer board horizontal. For the LQFP, the board meets the JESD51-3 specification. For the MAPBGA, the board meets the JESD51-9 specification.
3. Determined according to JEDEC Standard JESD51-6, Integrated Circuits Thermal Test Method Environmental Conditions—Forced Convection (Moving Air) with the board horizontal. For the LQFP, the board meets the JESD51-7 specification.
4. Determined according to JEDEC Standard JESD51-8, Integrated Circuit Thermal Test Method Environmental Conditions—Junction-to-Board. Board temperature is measured on the top surface of the board near the package.
5. Determined according to Method 1012.1 of MIL-STD 883, Test Method Standard, Microcircuits, with the cold plate temperature used for the case temperature. The value includes the thermal resistance of the interface material between the top of the package and the cold plate.
6. Determined according to JEDEC Standard JESD51-2, Integrated Circuits Thermal Test Method Environmental Conditions—Natural Convection (Still Air).

Table 21 Thermal Resistance Characteristics

3.5 Recommended Voltage and Current Operating Conditions

Recommended Supply Voltage	
Operating Voltage	1.8-V Core

Table 22 Normal Operating Conditions

Typical values assume you meet the following conditions (or other conditions as specified):

Symbol	Description	Min.	Max.	Unit
VDD	Digital supply voltage	-0.3	3.8	V
I_{DD}	Digital supply current	—	185	mA
V_{DIO}	Digital input voltage (except RESET_n, EXTAL, and XTAL)	-0.3	5.5	V
V_{AIO}	Analog1, RESET_n, EXTAL, and XTAL input voltage	-0.3	V_{DD} + 0.3	V
I_D	Maximum current single pin limit (applies to all port pins)	-25	25	mA
VDDA	Analog supply voltage	VDD - 0.3	VDD + 0.3	V

Table 23 Voltage and Current Operating Ratings

1. Analog pins are defined as pins that do not have an associated general-purpose I/O port function.

3.5.1 Maximum Current Consumption

Maximum Current			
Power Type	Condition	Value	Unit
Maximum Current from digital supply	VDD = 1.8 V @ 25°C	34	mA
Maximum Current from digital supply	VDD = 3.3 V @ 25°C	34	mA
Maximum Current from digital supply	VDD = 3.3 V @ 125°C	39	mA
Maximum Current from analog supply	VDDA = 1.8 V	4	mA
Maximum Current from analog supply	VDDA = 3.3 V	4	mA

Table 24 Maximum Current Rating

3.6 AC electrical characteristics

Unless otherwise specified, propagation delays are measured from the 50% to the 50% point, and rise and fall times are measured at the 20% and 80% points, as shown in the following figure.

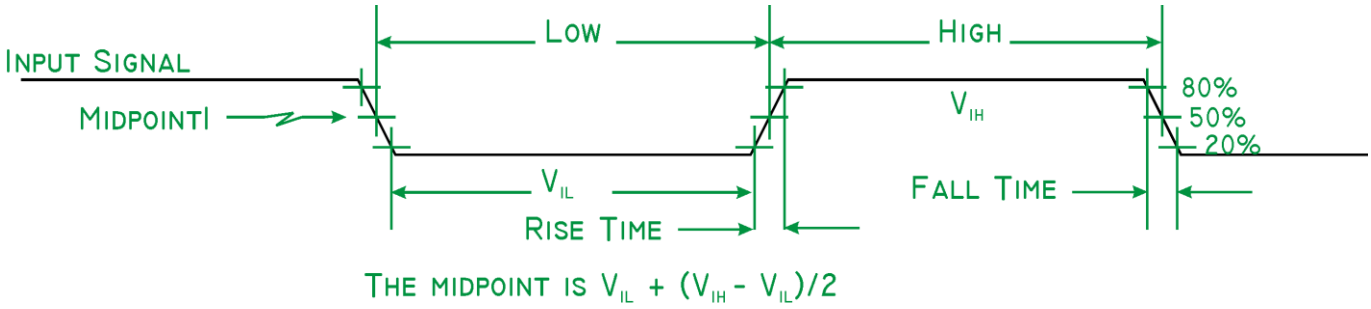


Figure 21 Input signal measurement reference

3.6.1 Voltage and Current Operating Requirements

Symbol	Description	Min.	Max.	Unit	Notes
V_{DD}	Supply voltage	1.71	3.6	V	
V_{DDA}	Analog supply voltage	1.71	3.6	V	
V_{DD}–V_{DDA}	V_{DD}-to-V_{DDA} differential voltage	–0.1	0.1	V	
V_{SS}–V_{SSA}	V_{SS}-to-V_{SSA} differential voltage	–0.1	0.1	V	
V_{IH}	Input high voltage when 2.7 V ≤V_{DD} ≤3.6 V when 1.7 V ≤V_{DD} ≤2.7 V	0.7 × V_{DD} 0.75×V_{DD}		V	
V_{IL}	Input low voltage when 2.7 V ≤V_{DD} ≤3.6 V when 1.7 V ≤V_{DD} ≤2.7 V		0.35 ×V_{DD} 0.3 × V_{DD}		
V_{HYS}	Input hysteresis	0.06 × VDD	—	V	
I_{ICDIO}	Digital pin negative DC injection current — single pin •V_{IN} < V_{SS}-0.3V	-5	—	mA	1
I_{ICAIO}	Analog2, EXTAL, and XTAL pin DC injection current single pin (See Note 2) V_{IN} < V_{SS}-0.3V (Negative current injection) V_{IN} > V_{DD}+0.3V (Positive current injection)	-5	+5	mA	3
I_{ICcont}	Contiguous pin DC injection current —regional limit, includes sum of negative injection currents or sum of positive injection currents of 16 contiguous pins • Negative current injection •Positive Current injection	-25	+25	mA	

NOTES:
1. All 5 V tolerant digital I/O pins are internally clamped to VSS through a ESD protection diode. There is no diode connection to V_{DD}. If VIN greater than V_{DIO}_MIN (=VSS-0.3V) is observed, then there is no need to provide current limiting resistors at the pads. If this limit cannot be observed then a current limiting resistor is required. The negative DC injection current limiting resistor is calculated as R=(V_{DIO}_MIN-VIN)/|IIC|.
2. Analog pins are defined as pins that do not have an associated general-purpose I/O port function.
3. All analog pins are internally clamped to V_{SS} and V_{DD} through ESD protection diodes. If VIN is greater than V_{AIO}_MIN (=V_{SS}-0.3V) and V_{IN} is less than V_{AIO}_MAX(=V_{DD}+0.3V) is observed, then there is no need to provide current limiting resistors at the

pads. If these limits cannot be observed then a current limiting resistor is required. The negative DC injection current limiting resistor is calculated as R=(V_{AIO}_MIN-VIN)/|IIC|. The positive injection current limiting resistor is calculated as R=(V_{IN}-V_{AIO}_MAX)/|IIC|. Select the larger of these two calculated resistances.

Table 25 Voltage and Current Operating Requirements

3.6.2 Voltage and Current Operating Behaviors

Symbol	Parameter	Min	Nom	Max	Unit	Notes
V_{OH}	Output high voltage • 2.7 V ≤ VDD ≤ 3.6 V, IOH = -2mA • 1.71 V ≤ VDD ≤ 2.7 V, IOH = -0.6mA	V_{DD} – 0.5 V_{DD} – 0.5	-	- -	V V	
I_{OHT}	Output high current total for all ports	-		100	mA	
V_{OL}	Output low voltage — low drive strength • 2.7 V ≤ VDD ≤ 3.6 V, IOL = 2mA • 1.71 V ≤ VDD ≤ 2.7 V, IOL = 0.6mA	- -	-	0.5 0.5	V V	
I_{OLT}	Output low current total for all ports	-	-	100	mA	
I_{IN}	Input leakage current (per pin)	-	-	1	µA	1
I_{OZ}	Hi-Z (off-state) leakage current (per pin)	-	-	1	µA	
R_{PU}	Internal pullup resistors	20	-	50	kΩ	2
R_{PD}	Internal pulldown resistors	20	-	50	kΩ	3
NOTES: 1. Measured at V_{DD}=3.6V 2. Measured at V_{DD} supply voltage = V_{DD} min and Vinput = V_{SS} 3. Measured at V_{DD} supply voltage = V_{DD} min and Vinput = V_{DD}						

Table 26 Voltage and Current Operating Behaviors

3.7 Reset Timing

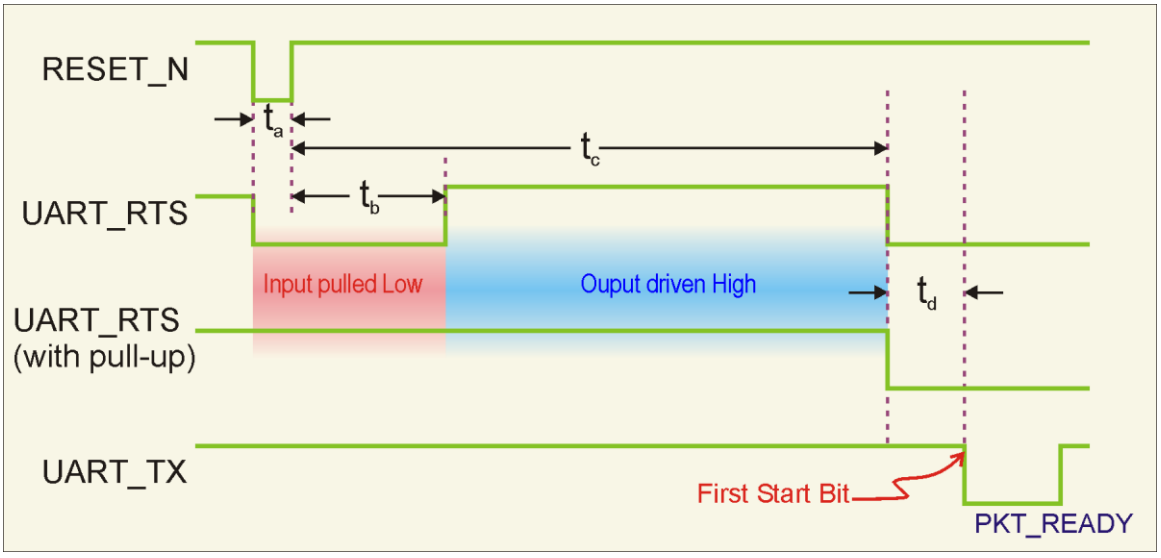


Figure 22 Reset Timing

Time	Minimum	Typical	Maximum	Notes
t_a	100 ns	-	-	reset pulse low time

t_b	1.7 ms	2.1 ms	2.5 ms	rising edge of RESET_N to rising edge of UART_RTS
t_c	42 ms	44 ms	46 ms	rising edge of RESET_N to falling edge of UART_RTS
t_d	70 µs	80 µs	90 µs	falling edge of UART_RTS to falling edge of UART_TX

Table 27 Reset Timing

3.8 Wake Timing

Time required to wake from Sleep Mode[1]			
Min	**Typ**	**Max**	**Notes**
2.5 ms	2.8 ms	3.0 ms	Falling edge of UART_RX to Falling edge of UART_TX.
2.5 ms	2.8 ms	3.0 ms	Rising or falling edge of UART_CTS to Falling edge of UART_TX.

1. Two methods exist to wake from Sleep Mode: either the receipt of a wake character or any transition of the UART_CTS pin. After the wake event occurs, a packet containing a PKT_READY field is sent in response (similar to RESET). Upon receiving PKT_READY, the device has exited Sleep Mode and has returned to Packet Mode.

Table 28 Wake Timing

3.9 Power Mode States and Transitions

The AMBE-4020™ Vocoder Chip has various power states as shown below.

Power Mode States		Worst case power consumption [1]	Description
Packet Mode	Encode	< 25.5 mW	In this power state, power consumption is variable depending upon packet load. The CPU clock is throttled to reduce power during periods of no packet activity. The chip can process configuration control packets, speech packets, or channel packets.
	Decode	< 16.8 mW	
	Full Duplex	< 31.2 mW	
Packet Mode - Idle		< 8 mW	In this power state, power consumption is static. The AMBE-4020™ has processed all packets in its queue and is awaiting the receipt of the next packet. This power state is entered automatically in Packet Mode when all packets have been processed. Upon receiving a packet, the AMBE-4020™ will automatically re-enter Packet Mode.
Low Power Packet Mode		< 4 mW	In this power state, power consumption is reduced, relative to Packet Mode, but packet throughput is also reduced. The maximum baud rate in this mode is 125000 baud. It is still possible to process channel and speech packets, but only at a rate much slower than real-time. This state provides a simple way of conserving power during periods of no activity. Prior to entering this state, the baud rate must be reduced to 125K baud or lower. Entry into this power state is achieved by sending { PKT_PMODE, $(PMODE)^8$ } where $(PMODE)^8 = 3$.

		Exiting this state is achieved by simply sending { PKT_PMODE, $(PMODE)^8$ } where $0 \leq (PMODE)^8 \leq 2$.
Low Power Packet Mode - Idle	< 2.2 mW	In this power state, power consumption is static. The AMBE-4020™ has processed all packets in its queue and is awaiting the receipt of the next packet. This power state is entered automatically in Low Power Packet Mode when all packets have been processed. Upon receiving a packet, the AMBE-4020™ will automatically re-enter Low Power Packet Mode.
Encoder Codec Mode	< 30.0 mW	In this power state, the encoder is functional and produces channel packets every 20 ms. The CPU clock is throttled to reduce power during encoder operation.
Decoder Codec Mode	< 18.6 mW	In this power state, the decoder is functional and expects to receive channel packets every 20 ms. The CPU clock is throttled to reduce power during decoder operation.
Full Duplex Codec Mode	< 37.4 mW	In this power state, both the encoder and decoder are functional. The encoder produces a channel packet every 20 ms and the decoder expects to receive a channel packet every 20 ms. The CPU clock is throttled to reduce power.
PTT Codec Mode - Idle	< 8 mW	In this power state, push-to-talk codec mode is enabled but neither the encoder or decoder is functioning because the ENC and DEC pins are both low. This idle mode is entered when both ENC and DEC are low and $(PMODE)^8 = 0$. This state is also entered temporarily during transition to other power states.
PTT Codec Mode - Encode	< 30.0 mW	In this power state, push-to-talk codec mode is enabled and the encoder is functioning because the ENC pin is held high.
PTT Codec Mode - Decode	< 18.6 mW	In this power state, push-to-talk codec mode is enabled and the decoder is functioning because the DEC pin is held high and the ENC pin is held low.
PTT Codec Mode - Low Power	< 2.2 mW	In this power state, push-to-talk codec mode is enabled but neither the encoder or decoder is functioning because the ENC and DEC pins are both low. This idle mode is entered when both ENC and DEC are low and $(PMODE)^8 = 1$. This state is also entered temporarily during transition to other power states. Power consumption is lower than Codec Mode – PTT Idle.
PTT Codec Mode - Sleep	< 9 µW	In this power state, push-to-talk codec mode is enabled but neither the encoder or decoder is functioning because the ENC and DEC pins are both low. This idle mode is entered when both ENC and DEC are low and $(PMODE)^8 = 2$. Power consumption is lower than Codec Mode – PTT Idle.
Halt Mode	< 4 µW	This is the lowest power state. The chip is unresponsive to any packet, and cannot function. Hardware reset is required to exit this state. The only lower power state would be to completely cut off power supply from the chip.
Sleep Mode	< 9 µW	In this power state, power is lower than Packet Mode Low Power. The chip cannot accept any normal packets, however all configuration is retained. To exit this state there are two options: 1. send a wake byte via PKT_TX_DATA. 2. any transition on PKT_CTS. In response to either wakeup method, the chip responds by sending a PKT_READY packet. The advantage of this power state over Halt is that configuration is retained. The

		time from wake to PKT_READY is much lower than from reset to PKT_READY.
Notes: [1] VDD = VDDA = 1.8V, 115200 baud, worst case vocoder rate when applicable, worst case codec interface selected (DMIC/DAC), echo canceller disabled.		

Table 29 Power Mode States

The following figure shows the power mode transitions. Any reset always brings the chip back to the normal run state.

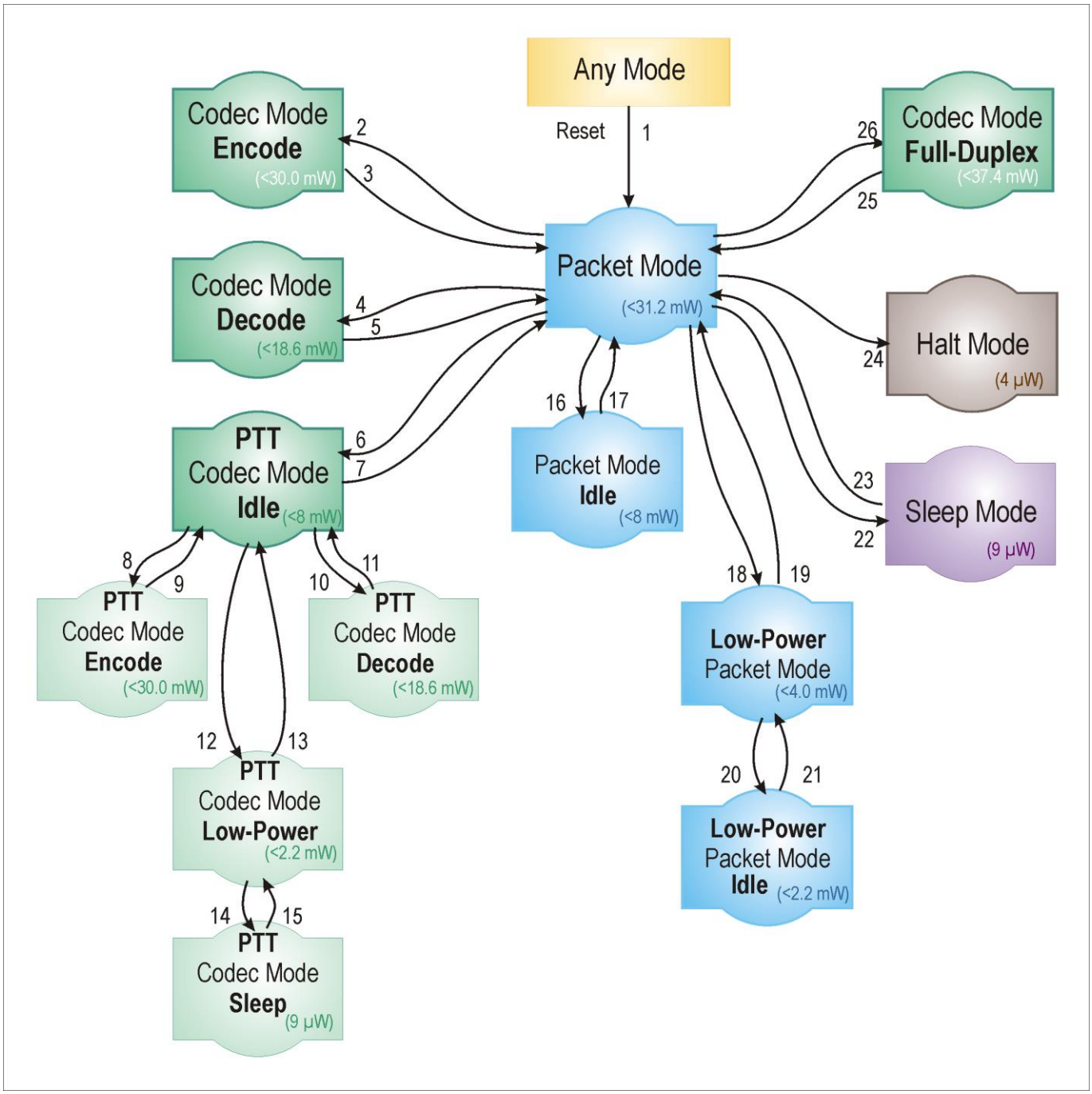


Figure 23 Power mode state transition diagram

Transition Number	Power Mode State
1	Hardware Reset
2	PKT_STARTCODEC with DUPLEX=DUPLEX_ENC=2 or $(MODE)^8 = 1$ and $(CODECCFG)^8 >> 6$ = DUPLEX_ENC after reset (as selected by boot configuration).
3	PKT_STOPCODEC
4	PKT_STARTCODEC with DUPLEX=DUPLEX_DEC=1 or $(MODE)^8 = 1$ and $(CODECCFG)^8 >> 6$ = DUPLEX_DEC after reset (as selected by boot configuration).
5	PKT_STOPCODEC
6	PKT_STARTCODEC with DUPLEX=DUPLEX_PTT=0 or $(MODE)^8 = 1$ and $(CODECCFG)^8 >> 6$ = DUPLEX_PTT after reset (as selected by boot configuration).
7	PKT_STOPCODEC
8	ENC=1
9	ENC=0
10	ENC=0 and DEC =1
11	DEC=0 or ENC=1
12	ENC=0 and DEC=0 and (PMODE=1or PMODE=2)
13	ENC=1 or DEC=1
14	ENC=0 and DEC=0 and PMODE=2
15	ENC=1 or DEC=1
16	receive packet queue is empty
17	packet is received
18	PKT_PMODE (3) or $(PMODE)^8 = 3$ after reset (as selected by boot configuration).
19	PKT_PMODE (0,1,2)
20	receive packet queue is empty
21	packet is received
22	PKT_PMODE (4) or $(PMODE)^8 = 4$ after reset (as selected by boot configuration).
23	Wake Byte or UART_CTS Transition
24	PKT_PMODE (5) or $(PMODE)^8 = 5$ after reset (as selected by boot configuration).
25	PKT_STOPCODEC
26	PKT_STARTCODEC with DUPLEX=DUPLEX_FULL=3 or $(MODE)^8 = 1$ and $(CODECCFG)^8 >> 6$ = DUPLEX_FULL after reset (as selected by boot configuration).

Table 30 Power mode state transitions

3.10 Codec Mode Power Consumption

During codec mode, the AMBE-4020™ power consumption depends primarily on the following five factors:

1. Supply Voltage. The AMBE-4020™ power consumption is dependent upon the supply voltage level. The recommended supply voltage level is 1.8 V. If power consumption is not a concern then a higher supply voltage may be used.

2. Duplex. The encoder consumes more power than the decoder does. Compare Figure 24 Power Consumption and Figure 25 .

3. Vocoder rate: Power consumption varies depending upon which vocoder bit rate is selected.

In general encoder power consumption is lowest for low bit-rates and highest for high bit rates. Encoder rate 31 (2000 bps) uses the least power whereas encoder rate 21(9600 bps) uses the most power. See Figure 24 Power Consumption . All other encoder rates use power between these two extremes.

In general, decoder power consumption is lower for low bit rates that do not utilize a lot of FEC or soft-decision decoding. In general, decoder power consumption is higher for rates that have a higher bit rate or rates that utilize more FEC with soft-decision decoding. Decoder rate 31 (2000 bps voice + 0 bps FEC) uses the least power whereas decoder rate 60 (4000 bps voice + 4000 bps FEC) uses the most power. See Figure 25 . All other decoder rates consume power between these two extremes.

4. Speech interface selection: ADC, DMIC, or I2S for encode or DAC or I2S for decode.

For the encoder for any selected rate, the I2S interface consumes the least power, the ADC consumes intermediate power, and the DMIC consumes the most power. See Figure 24 Power Consumption .

For the decoder for any selected rate, the I2S interface consumes less power than the DAC interface. See Figure 25 .

5. Baud rate selected for the UART interface: Power is also dependent upon the chosen UART baud rate.

In general, the lowest power is consumed for baud rates ≤ 125000. For baud rates > 125000 but ≤ 250000 power consumption increases. For baud rates > 250000 baud power consumption again increases. See Figure 28 Codec Mode Power Variation with Baud Rate. The default baud rate is 115200 and typically offers the best power vs bandwidth vs compatibility compromise. Typically, in codec mode there will not be any need for higher baud rates since the bandwidth of the channel data is relatively small. Reducing baud rate below 125000, in general will not save any power, however it may be useful to do so due to requirements of the connected hardware.

Also, note that there are four different code modes: Encoder Codec Mode, Decoder Codec Mode, Full Duplex Codec Mode and PTT Codec Mode. PTT Codec Mode offers additional power benefits, since the AMBE-4020™ can automatically start the encoder when requested by the ENC pin being high. Similarly, the decoder automatically starts when the DEC pin is high (only if the ENC pin is low). The AMBE-4020™ offers three different levels of power savings when both the ENC and DEC pins are low. Generally the most aggressive power savings ($(PMODE)^8 = 2$) is the best choice.

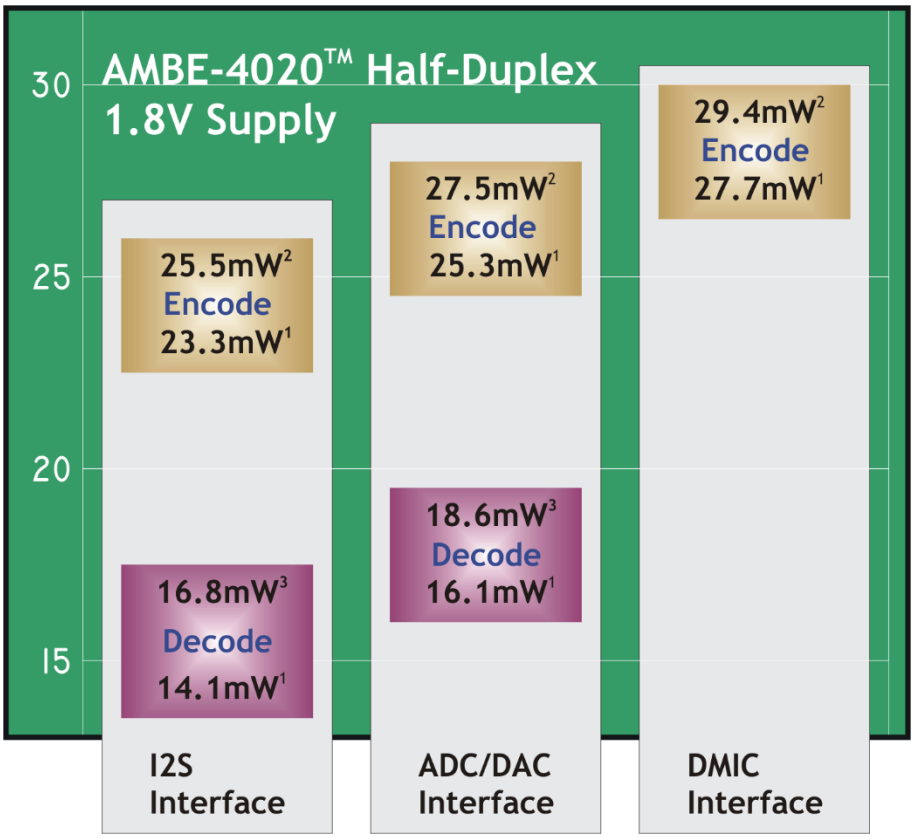


Figure 24 Power Consumption Half-Duplex (1.8V Supply)

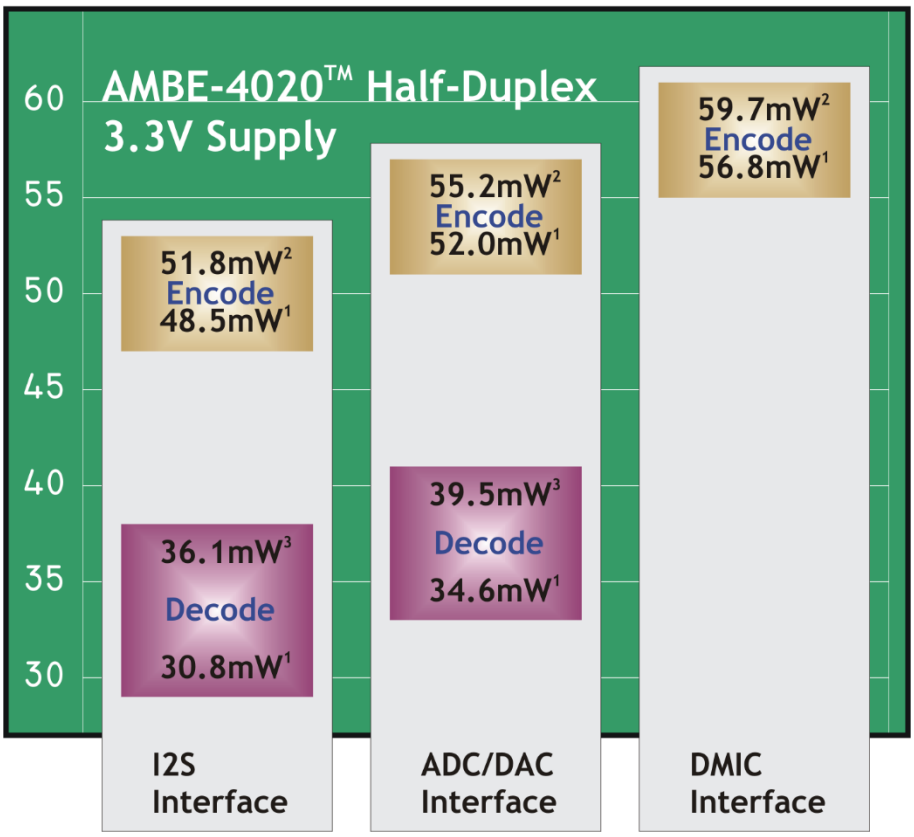


Figure 25 Power Consumption Half-Duplex (3.3V Supply)

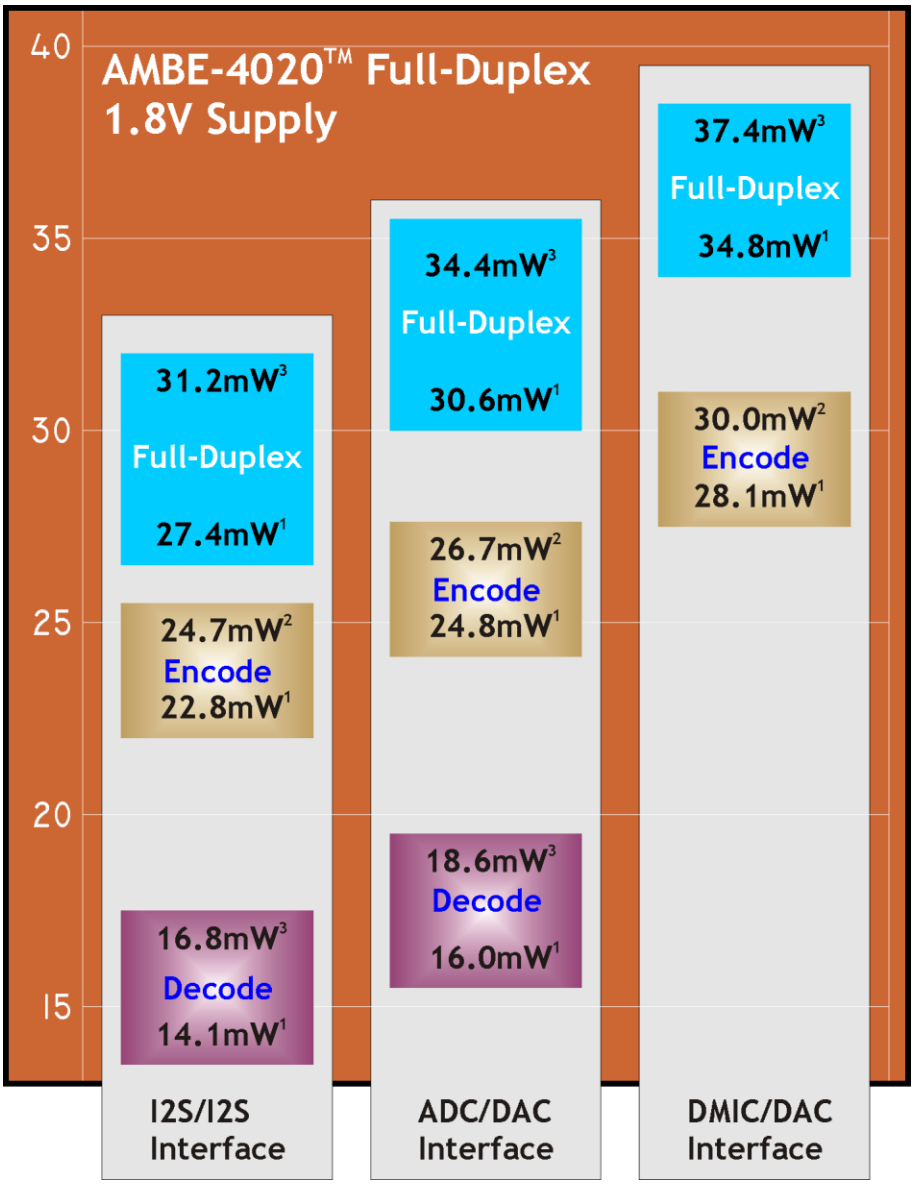


Figure 26 Power Consumption Full-Duplex (1.8V Supply)

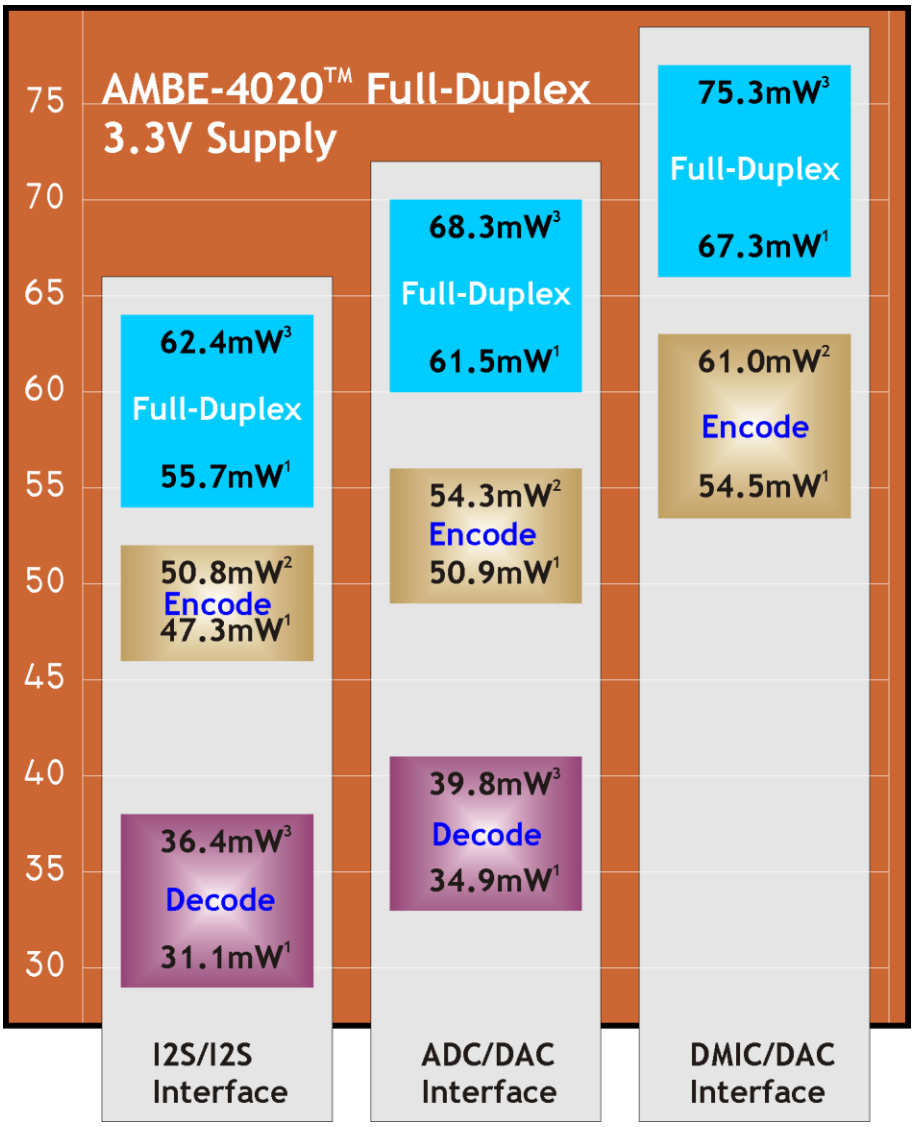


Figure 27 Power Consumption Full-Duplex (3.3V Supply)

NOTES:

1. The Index rate with the lowest power consumption requirement is Rate 31 (2000bps)
2. The Encoder index rate with the highest power consumption requirement is Rate 21 (9600bps)
3. The Decoder index rate and the Full-Duplex index rate with the highest power consumption requirement is Rate 60 (9600bps)

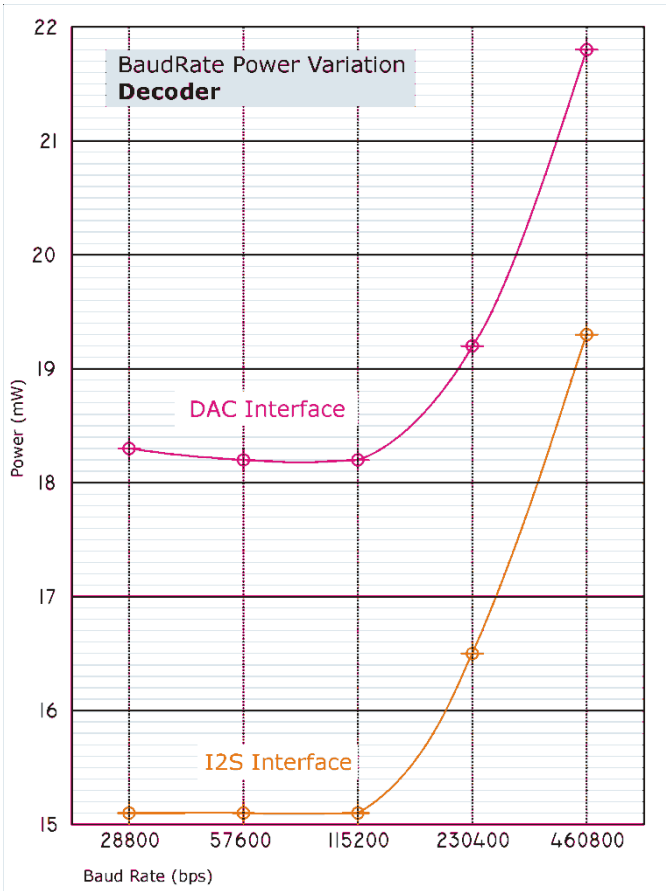


Figure 28 Codec Mode Power Variation with Baud Rate

The following tables compare the current consumption by the AMBE-4020™ Half-Duplex and AMBE-4020™ Full-Duplex when operating in encoder codec mode and decoder codec mode.

AMBE-4020™ Codec Mode Typical Power Consumption[1,2,3]						
Conditions Tested			AMBE-4020™ Half-Duplex Power (mW)		AMBE-4020™ Full-Duplex Power (mW)	
Duplex[4]	Vocoder Rate[5]	Interface(s)	@ 1.8V Supply	@ 3.3V Supply	@ 1.8V Supply	@ 3.3V Supply
Full	31/best	I2S/I2S	Not Supported		27.4	55.7
Full	31/best	ADC/DAC			30.6	61.5
Full	31/best	DMIC/DAC			34.8	67.3
Full	60/worst	I2S/I2S			31.2	62.4
Full	60/worst	ADC/DAC			34.4	68.3
Full	60/worst	DMIC/DAC			37.4	75.3
Encode	31/best	I2S	23.3	48.5	22.8	47.3
Encode	31/best	ADC	25.3	52.0	24.8	50.9
Encode	31/best	DMIC	27.7	56.8	28.1	54.5
Encode	21/worst	I2S	25.5	51.8	24.7	50.8
Encode	21/worst	ADC	27.5	55.2	26.7	54.3
Encode	21/worst	DMIC	29.4	59.7	30.0	61.0
Decode	31/best	I2S	14.1	30.8	14.1	31.1
Decode	31/best	DAC	16.1	34.6	16.0	34.9
Decode	60/worst	I2S	16.8	36.1	16.8	36.4
Decode	60/worst	DAC	18.6	39.5	18.6	39.8

NOTES:
1. Current from VDD only
2. Current is mostly periodic with a period of 20 ms. Typical power is estimated by measuring the average current over 650 separate 20 ms periods.
3. Measured using 115200 baud with echo canceller and echo suppressor disabled.
4. Duplex=Encode applies to either Encoder Codec Mode or PTT Codec Mode – Encode. Duplex=Decode applies to either Decoder Codec Mode or PTT Codec Mode – Decode. Duplex=Full applies to Full Duplex Codec Mode
5. Power varies with the vocoder rate selected. For Duplex=Full, the best case rate is rate 31 and the worst case rate is rate 60. For Duplex=Encode, the best case rate is 31 and the worst case rate is 21. For Duplex=Decode, the best case rate is 31 and the worst case rate is 60. For other vocoder rates, the power consumption falls in between the best case and the worst case.

Table 31 Typical Consumption from VDD during Codec Mode

Typical Codec Mode Power Consumption Vs Baud Rate[1,2]			
Baud Rate	Duplex	AMBE-4020™ Half-Duplex Power (mW)	AMBE-4020™ Full-Duplex Current (mW)
28800	encode	23.8	24.0
	decode	14.0	15.2
57600	encode	23.7	23.2
	decode	14.1	14.3
115200	encode	23.3	22.8
	decode	14.1	14.1
230400	encode	25.0	24.3
	decode	15.5	15.5
460800	encode	27.5	28.2
	decode	18.2	20.0

Notes:
1. Current from VDD only
2. VDD=1.8V, Interface=I2S, Vocoder Rate 31

Table 32 Codec Mode Power Variation with Baud Rate

Additional Power Consumption from VDD when Echo Canceller is Enabled[1,2]	
(ECHOLEN)[8]	Additional Power Attributed to Echo Canceller
8	1.62 mW
16	1.80 mW
…	$1.44 + 0.0225 \times (\text{ECHOLEN})^8$ mW
120	4.14 mW
128	4.32 mW

Notes:
1. Estimated typical power fit to measurement data
2. Echo Canceller is only supported by the AMBE-4020™ Full-Duplex in Full Duplex Codec Mode.

Table 33 Additional Power Consumption from VDD by Echo Canceller

3.11 Packet Mode Power Consumption

During packet mode, power consumption depends primarily on the following five factors:

1. Supply Voltage: For packet mode, as with codec mode, 1.8 V supply voltage is recommended for lowest power consumption.

2. Duplex: For packet mode, as with codec mode, the encoder consumes more power than the decoder.

3. Vocoder rate: Power consumption in packet mode also varies depending upon which vocoder bit rate is selected.

4. Packet Frequency: Power consumption is dependent upon how frequently channel packets or speech packets are received. In general, for a real-time system the AMBE-4020™ should receive one speech packet or one channel packet at 20 ms intervals. The AMBE-4020™ can tolerate a higher packet frequency but that increases power consumption proportionally. A lower packet frequency reduces power consumption

5. Baud rate selected for the UART interface: Power is also dependent upon the chosen UART baud rate. The minimum recommended baud rate for packet mode is 172800 baud. Packet mode requires higher baud rates than codec mode since the UART must transfer both speech data and channel data. Baud rates ≥ 172800 baud ≤ 250000 baud provide optimal power. Higher baud rates consume more power but can increase throughput and reduce transmission delay.

For packet mode, typical power consumption for a given duplex, vocoder rate, and baud rate is the same as it is for codec mode when the I2S interface is selected. This assumes that the packet rate is one Speech/Channel packet per 20 ms (on average).

In packet mode, if the AMBE-4020™ stops receiving packets it automatically will switch into the Packet Mode Idle state that consumes less power. Further power savings are possible by sending PKT_PMODE to switch into a lower power state.

$(PMODE)^8 = 3$: The AMBE-4020™ transitions from packet mode to Low Power Packet Mode. In low power packet mode, the AMBE-4020™ can still receive packets and generate response packets, but the response time for packets is reduced. The AMBE-4020™ can even process speech/channel packets in this mode but at a rate much less than real time. When no packets are received in Low Power Packet Mode, the AMBE-4020™ will automatically switch to Low Power Packet Idle Mode.

$(PMODE)^8 = 4$: The AMBE-4020™ transitions from Packet Mode to Sleep Mode, which is a very low power state. Sleep mode is useful to conserve power during periods of no activity, yet maintaining all settings upon exit from this mode. There are two different ways of waking from sleep mode. The first method is to send a wake byte. The second method is to transition the UART_CTS pin. After waking via either method, the AMBE-4020™ transmits a packet containing PKT_READY. Following the receipt of PKT_READY the AMBE-4020™ has reentered Packet Mode and may begin receiving packets and transmitting response packets once again.

$(PMODE)^8 = 5$: The AMBE-4020™ transitions from Packet Mode to Halt Mode, which is the lowest power state. However, the only way to exit this lowest power state is via a hard-reset. After entering this state, the AMBE-4020™ does not respond to any UART packets.

Power Consumption During Power Savings Modes[1,2]			
Mode	Notes	Power Consumption	
		VDD=1.8V	VDD=3.3V
Packet Mode - Idle	Packet Mode when no packets are received or transmitted	8.0 mW	19.8 mW
PTT Codec Mode - Idle	PTT Codec Mode when: ENC=DEC=0 $(PMODE)^8 = 0$		
Low Power Packet Mode - Idle	Low Power Packet Mode when no packets are received or transmitted. $(PMODE)^8 = 3$	2.2 mW	9.6 mW
PTT Codec Mode - Low Power	PTT Codec Mode when: ENC = DEC = 0 $(PMODE)^8 = 1$		

Sleep Mode	$(PMODE)^8 = 4$	9 µW	16 µW
PTT Codec Mode – Sleep	PTT Codec Mode when: ENC=DEC=0 $(PMODE)^8 = 2$		
Halt Mode	$(PMODE)^8 = 5$	4 µW	6 µW

Notes:
1. All measurements include combined power from both VDD and VDDA
2. Measurements at 25°C, leakage currents increase at higher temperatures.
3. AMBE-4020™ Full-Duplex idles at 6.3 mW

Table 34 Power Consumption from VDD and VDDA during Power Saving Modes

Power Consumed from VDDA[1,2]					
ADC/DAC Used?[3]		AMBE-4020 Half-Duplex		AMBE-4020 Full-Duplex	
ADC	DAC	VDD=VDDA=1.8V	VDD=VDDA=3.3V	VDD=VDDA=1.8V	VDD=VDDA=3.3V
		< 4 µW	< 6 µW	< 4 µW	< 6 µW
	☑	1.23 mW	2.41 mW	0.56 mW	1.06 mW
☑		0.98 mW	1.92 mW	0.76 mW	1.49 mW
☑	☑	Not Supported		1.01 mW	2.12 mW

Notes:
1. Power is consumed from VDDA only during Codec Mode when the ADC and/or DAC interface(s) are selected.
2. VREF_OUT not connected.
3. Any mode that does not utilize ADC/DAC only leakage power is consumed. This includes Packet Mode, Codec Mode, and PTT Codec Mode.

Table 35 Power Consumption from VDDA

4 Initial Design Considerations

4

Some of the initial design considerations the application engineer will face are the following:

1. Full-Duplex or Half-Duplex communication system
2. Speech and FEC rates (2000 – 9600 bps)
3. Mode of operation (codec mode or packet mode)
4. Codec interfaces: (*for codec mode only!*)
- Input: I2S, ADC or Digital Mic
- Output: I2S or DAC
5. Choice of using an internal or external A/D-D/A

Integrating the AMBE-4020™ Vocoder Chip into a communication system requires the selection of various components. The AMBE-4020™ Vocoder Chip offers multiple interfaces for flexibility in integration into a variety of design configurations.

In its simplest model, the AMBE-4020™ Vocoder Chip can be viewed as two separate components, the Encoder and the Decoder. The Encoder receives an 8 kHz sampled stream of speech data (16-bit linear, 8-bit A-law, or 8-bit µ-law) and outputs a stream of channel data at the desired rate. Simultaneously, the AMBE-4020™ Vocoder Chip receives compressed voice channel data. This data is decoded by the AMBE-4020™ Vocoder Chip, then reconstructed into a digital speech signal and sent to the D/A. The encoder and decoder functions are fully asynchronous.

The special functions of the AMBE-4020™ Vocoder Chip, such as voice activity /detection, power mode control, data/FEC rate selection, etc. can specified in a custom boot configuration or through the packet interface.

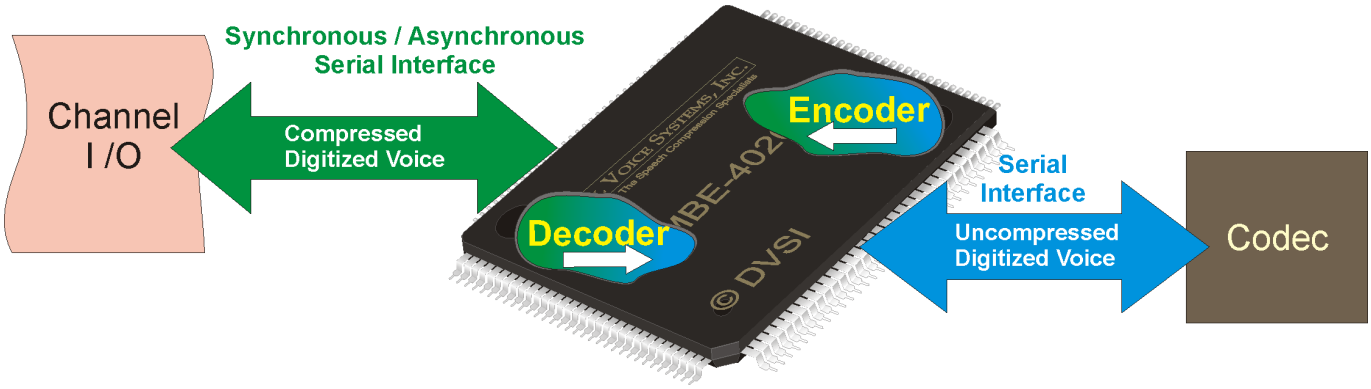


Figure 29 Basic Operation

4.1 Vocoder Speech and FEC Rate Selection

The voice-coding rate as well as the FEC coding rate can be selected individually on the AMBE-4020™ Vocoder Chip. These rates are selected by using a configuration control packet or by using BOOT0/BOOT1 to select a pre-programmed boot configuration. See Section 2.13:Boot Configuration Pins.

4.1.1 Vocoder Front End Requirements

In order to ensure proper performance from the voice coder, it is necessary for the vocoder front end to meet a set of minimum performance requirements. For the purposes of this section, the vocoder front end is considered to be the total combined response between microphone/speaker and the digital PCM interface to the vocoder, as shown in Figure 30 Typical Vocoder Implementation. This includes any analog electronics plus the A-to-D and D-to-A converters as well as any digital filtering performed prior to the voice encoder or after the voice decoder.

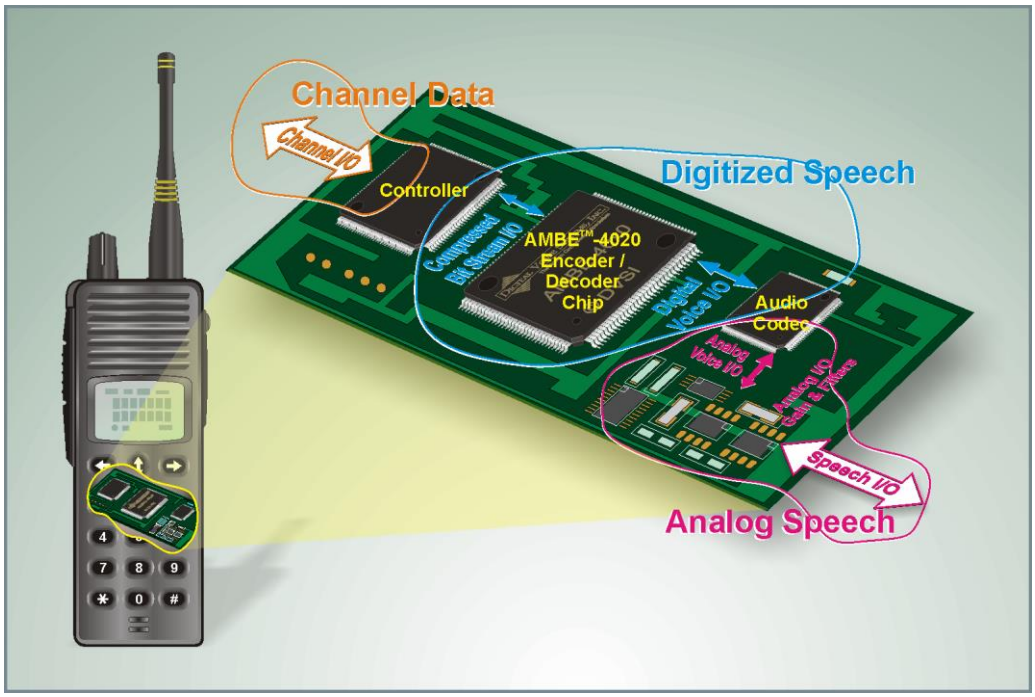


Figure 30 Typical Vocoder Implementation

By default, the AMBE+™ voice encoder and decoder operate with unity (i.e. 0 dB) gain. Consequently, the analog input and output gain elements shown in Figure 2 are only used to match the sensitivity of the microphone and speaker with the A-to-D

converters and D-to-A converters, respectively. It is recommended that the analog input gain be set such that the RMS speech level under nominal input conditions is 25 dB below the saturation point of the A-to-D converter (+3 dBm0). This level, which equates to
-22 dBm0, is designed to provide sufficient margin to prevent the peaks of the speech waveform from being clipped by the A-to-D converter.

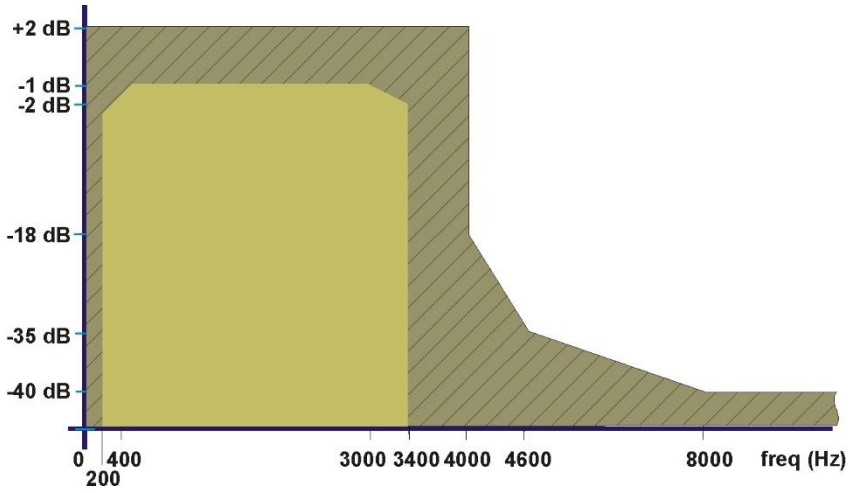


Figure 31 Front End Input Filter Mask

The voice coder interface requires the A-to-D and D-to-A converters to operate at an 8 kHz sampling rate (i.e. a sampling period of 125 microseconds) at the digital input/output reference points. This requirement necessitates the use of analog filters at both the input and output to eliminate any frequency components above the Nyquist frequency (4 kHz). The recommended input filter mask is shown in Figure 2 - C, and the recommended output filter mask is shown in Figure 2 - D. For proper operation, the shaded zone of the respective figure should bound the frequency response of the front-end input and output.

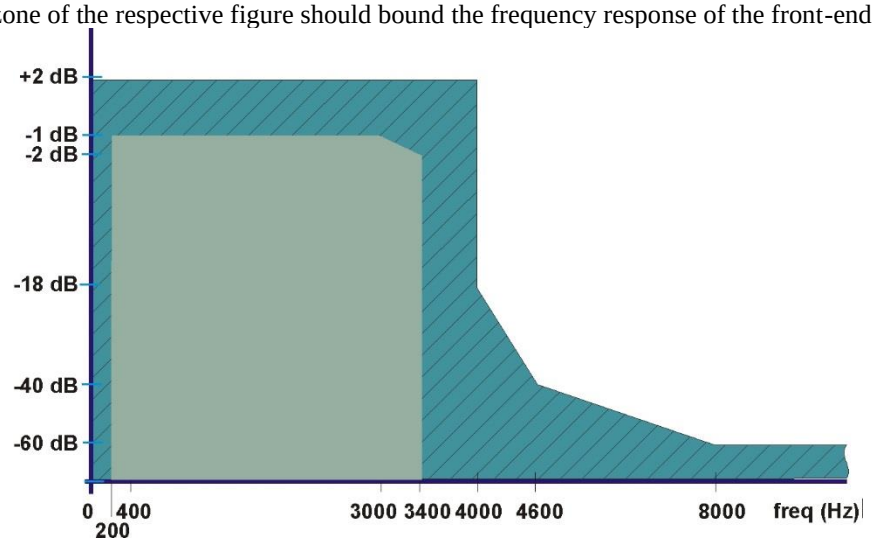


Figure 32 Front End Output Filter Mask

This document assumes that the A-to-D converter produces digital samples where the maximum digital input level (+3 dBm0) is defined to be +/- 32767, and similarly, that the maximum digital output level of the D-to-A converter occurs at the same digital level of +/- 32767. If a converter is used, which does not meet these assumptions then the digital gain elements shown in Figure 2 should be adjusted appropriately. Note that these assumptions are automatically satisfied if 16-bit linear A-to-D and D-to-A converters are used, in which case the digital gain elements should be set to unity gain.

An additional recommendation addresses the maximum noise level measured at the output reference points shown in Figure 2-B with the corresponding inputs set to zero. DVSI recommends that the noise level for both directions should not exceed -60 dBm0 with no corresponding input.

4.2 Codec Interface Selection

Basic communication to/from the AMBE-4020™ Vocoder Chip consists of input digitized speech data samples, output digitized speech data samples, input compressed speech data and output compressed speech data. The decision a designer has to make is what interface to use for codec input (I2S, ADC or Digital Mic In) and what interface to use for codec output (I2S or DAC).

NOTE: The Full duplex codec mode provided by theAMBE-4020™ Full-Duplex chip supports only the following combinations of codec interfaces: I2S input with I2S output, ADC input with DAC output, or DMIC input with DAC output. For example, selecting I2S input with DAC output is invalid. However using I2S input with DAC output is allowed for push-to-talk codec mode since the two interfaces are not used simultaneously.

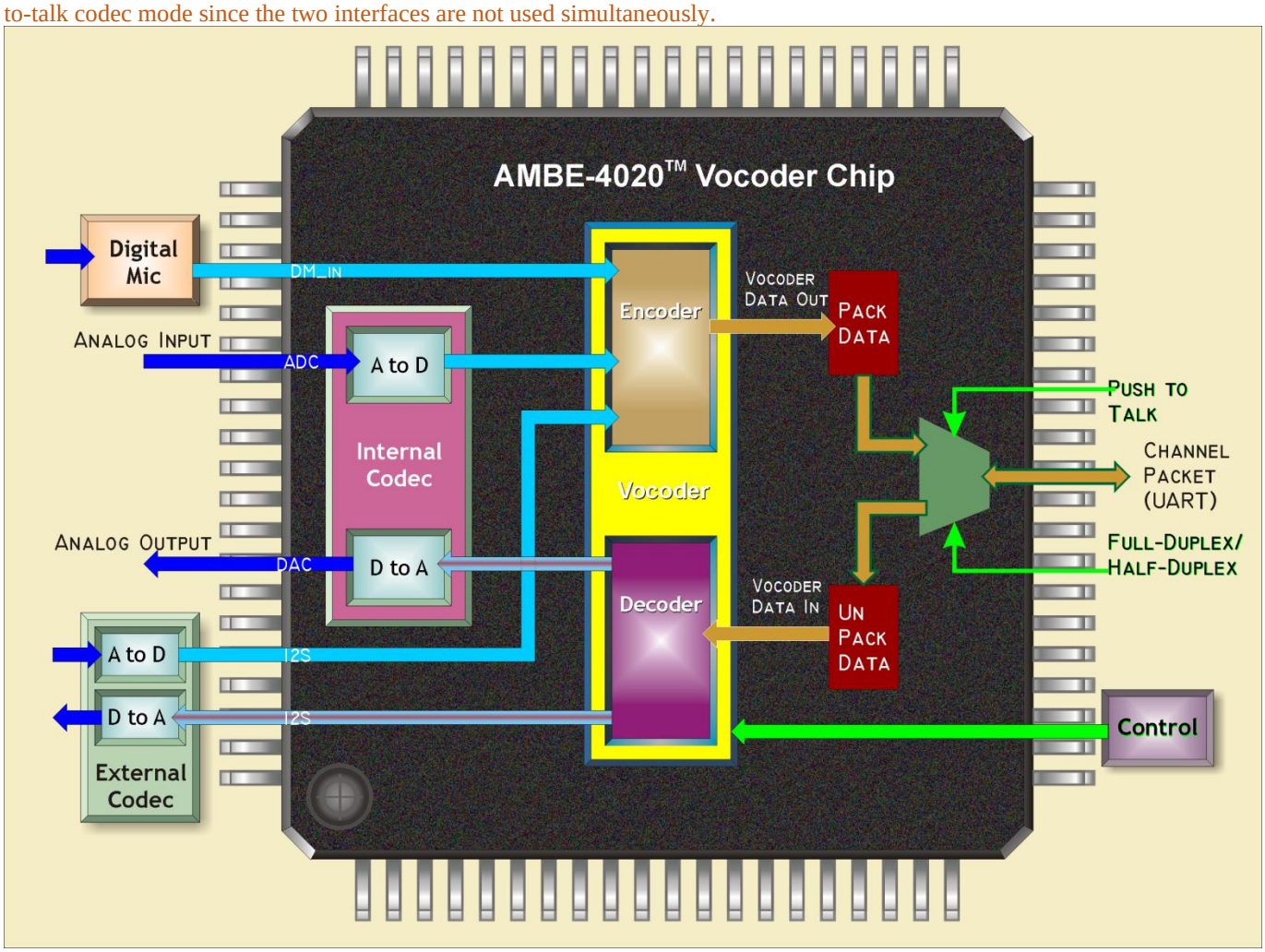


Figure 33 AMBE-4020™ Block Diagram

4.3 External A/D D/A selection

The AMBE-4020™ Vocoder Chip can be configured to transmit and receive digitized speech to and from most linear, a-law, or μ-law A/D-D/A codecs. The format of the incoming and outgoing speech data streams are coupled, that is to say they must be the same format (16-bit linear, 8-bit a-law, or 8-bit μ-law). The digitized speech from the external A/D is converted into compressed digital data (encoded) by the AMBE-4020™ Vocoder Chip and the channel data is output to the packet interface. Alternatively, speech data can be sent to/from the AMBE-4020™ Vocoder Chip via a packet interface.

The choice of the A/D-D/A chip is critical to designing a system with superior voice quality. Given that a-law and μ-law companding chips are already incorporating some compression to reduce the number of bits per sample, it is recommended that, when possible, a 16-bit linear device be used for maximum voice quality. When choosing a device, pay particular attention to signal to noise ratios and frequency responses of any filters that may be present on the analog front end of these chips. Generally speaking the flatter the frequency response over the voice spectrum (20-4000Hz) the better the overall system will sound. The a-law and μ-law interfaces are mainly provided for the design engineer who is trying to fit to pre-existing conditions or is under cost savings restraints.

4.4 Vocoder Features

The special functions of the AMBE-4020™ Vocoder Chip, such as voice activity detection, DTMF, data/FEC rate selection, power mode control, etc. can be controlled using configuration control packets sent after reset or by using the BOOT1/BOOT0 pins to select a pre-programmed boot configuration.

4.5 Voice Activity Detection & Comfort Noise Insertion

The Voice Activity Detection (VAD) algorithm along with the Comfort Noise Insertion (CNI) feature of the AMBE-4020™ Vocoder Chip performs useful functions in systems trying to convert periods of silence, that exist in normal conversation, to savings in system bandwidth or power. VAD and CNI can be enabled as part of a control packet.

With the VAD functions enabled, when periods of silence occur, the encoder will output a silence frame (in-band). This silence frame contains information regarding the level of background noise, which allows the corresponding decoder to synthesize a "Comfort Noise" signal at the other end. The comfort noise is intended to give the listener the feeling that the call is still connected, as opposed to producing absolute silence, which can give the impression that, the call has been "dropped". The decoder will produce a comfort noise frame if it receives an in-band silence frame (produced only by an encoder with VAD enabled). The synthesis of a Comfort Noise frame by the decoder is not dependent on VAD being enabled.

If the VAD features are being used to reduce transmit power during times of conversational silence, DVSI recommends that a silence frame be transmitted at the start of the period and approximately each 500-1000 milliseconds thereafter. This is to ensure that the parameters regarding the levels of background noise are transmitted to the decoder for the smoothest audible transitions between synthesized speech and synthesized silence.

The silence threshold value is -25 dBm0 in the VAD algorithm. Each frame that exceeds this level will be classified as voice. If the frame level is less than -25 dBm0 the voice/silence decision will be determined based upon various adaptive thresholds.

DTX is disabled by default after reset when boot configuration 0 is selected. The field PKT_ECONTROL may be used to enable or disable DTX.

4.6 DTMF Dual Tone Multiple Frequency, Detection and Generation

The AMBE-4020™ Vocoder Chip is capable of detecting, transmitting, and synthesizing DTMF tones, KNOX tones, call progress tones, and single frequency tones. When the encoder detects a tone, it passes the tone data in-band (within the regular

voice data bits) so that tones pass seamlessly from the encoder to the decoder for synthesis. The decoder synthesizes a tone in response to reception of an in-band tone frame.

The following text provides an outline of the tone features that are available. For specific details, the user must refer to the section that provides details on the packet formats.

When the encoder detects a tone, the output channel packet can optionally report the identity (frequency(s) and amplitude) of the detected tone. The TONEDET bits within a PKT_CHANFMT field specify whether tones detected by the encoder are reported out-of-band using PKT_TONEDET fields within outgoing channel packets. The PKT_TONEDET field identifies the frequency(s) and amplitude of the detected tone.

When the decoder receives a tone, it can optional report the identity of the received tone. The TONERCV bits within a PKT_CHANFMT field specify whether tones received in-band by the decoder are reported using PKT_TONERCV fields within outgoing channel packets. The PKT_TONERCV field identifies the frequency(s) and amplitude of the received tone.

For packet mode, the TONERCV bits within a PKT_SPCHFMT field specify whether tones received in-band by the decoder are reported using PKT_TONERCV fields within outgoing speech packets.

A PKT_TONEXMT field sent to the encoder within a channel packet is used to force the encoder to output in-band tone data. The frequency(s), amplitude, and duration of the tone are specified in the field.

A PKT_TONEGEN field sent to the decoder within a channel packet is used to force the decoder to synthesize a tone. The decoder will synthesize the tone regardless of what in-band signal is received. The frequency(s), amplitude, and duration of the tone are specified by the field.

Note that some backward compatible rates do not encode tones in-band in such a fashion that they can be identified as a tone by the decoder. All tones can be detected by the encoder and the identity of the tone can be reported by the PKT_TONEDET field. All tones can be generated at the decoder using the PKT_TONEGEN field even for rates that do not support in-band signaling for that tone type. The following table specifies for each vocoder rate index, whether the rate supports in-band signaling for each tone type. A ☑ indicates that tone signaling is supported. There is a tradeoff, in that increased backward compatibility equates to reduced in-band tone signaling support. In general, it is best to select AMBE+2 rates, unless backward compatibility is required.

Vocoder Rate Index		Backward Compatibility			In-band Tone Signaling Support			
		AMBE-1000	AMBE-2000	AMBE-3000	DTMF tones	single tones	call-progress	KNOX tones
0-15	AMBE	☑	☑	☑	☑			
22, 25, 29, 31, DSTAR,	AMBE		☑	☑	☑	☑	☑	
16-21, 23, 24, 26-28, 30, 32,	AMBE+		☑	☑	☑	☑	☑	
33-61	AMBE+2			☑	☑	☑	☑	☑

Table 36 In-band Tone Signaling Support

4.7 Soft Decision Error Correction

Significant improvement in FEC performance can be added by setting up a receiver so that the demodulator is making a finer estimation of the received energy prior to sending it to the decoder, this is called soft-decision decoding. To use Soft Decision Error Correction use the PKT_CHAND4 (ID 0x17) field in the channel packet. The AMBE-4020™ Vocoder Chip utilizes a 4-bit soft decision decoder. The bits are defined as follows:

Decision Value (Binary)	Interpretation
0000	Most confident 0
0111	…
1000	…

1111	Most confident 1

Table 37 Soft Decision Error Correction

The user must implement circuitry at the receive end of the channel for making a finer (4 bit) estimation of the received energy. The AMBE-4020™ Vocoder Chip uses a different channel data field (PKT_CHAND4) to specify channel data represented by 4 soft decision (SD) bits. The decoder will make the decision of whether or not a 1 or a 0 is represented by the soft-decision bits.

4.8 Skew Control

The AMBE-4020™ Vocoder Chip processes speech in voice frames that are approximately 20 ms in duration. Skew control can provide the designer with flexibility in dealing with clock drift. The AMBE-4020™ Vocoder Chip skew control feature allows the vocoder chip to compensate for drift between the frame and sample rate clocks.

Codec Mode
The **PKT_STARTCODEC** field selects whether or not skew control is enabled. When skew control is enabled, the AMBE-4020™ Vocoder Chip adjusts the frame boundaries so that they occur on the rising edge of the IFRAME signal. The user must supply the IFRAME signal such that the frame size varies between 156 and 164 samples. i.e. 48.75Hz ≤ IFRAME ≤ 51.25 Hz (50Hz ± 1.25Hz or 50 Hz ± 2.5%). For best quality, it is expected that the IFRAME signal is a stable framing signal such that there is not a lot of variation from frame to frame.

Packet Mode Skew Control Enable
In packet mode, the normal length of the input speech packets is 160 samples. However, this can vary between 156 and 164 samples in length. Output speech packets can also vary in length from 156 to 164 samples. When passing speech packets to the AMBE-4020™ encoder, the number of samples in each speech packet is specified within the PKT_SPEECHD field. When passing channel packets to the decoder, the number of samples to be output by the decoder is specified by a PKT_SAMPLES field within the channel packet. If the channel packet does not contain a PKT_SAMPLES field, then the number of samples output in the decoder's responding speech packet will default to 160. Note that the user should avoid too much variation in the number of speech samples. Normally the number of samples should be 160, with occasional adjustments when needed.

4.9 Noise Suppressor

The integrated Noise suppressor feature of the AMBE-4020™ Vocoder Chip is used to reduce the effect of background noise in the encoder input signal. The Noise suppressor is applied to both silence frames and voice frames, but not tone frames. When the noise suppressor is started, it may take up to a few seconds to converge allowing it to begin fully working.

The noise suppressor is automatically enabled after reset when boot configuration 0 is selected. The field PKT_ECONTROL may be used to enable or disable the noise suppressor.

4.10 Echo Canceller

The AMBE-4020™ Vocoder Chip's voice coder contains an echo canceller that can be selectively enabled or disabled via the EC_ENABLE bit in ECONTROL flags. See Table 40 ECONTROL Flags.

The echo canceller may only be enabled for full duplex codec mode. Echo cancellation is not supported in packet mode.

The echo canceller's filter length may be set by sending a control packet containing the PKT_ECHOLEN field prior to entering full duplex codec mode. This packet field sets a variable named $(ECHOLEN)^{16}$ anywhere between 8 and 128 samples, where $(ECHOLEN)^{16}$ must be evenly divisible by 8. $(ECHOLEN)^{16}$ / 8 is the filter length in milliseconds.

The INIT_EC bit in the PKT_INIT field can be used to initialize both the echo canceller and the echo suppressor. See Table 52 PKT_INIT Field

The packet field PKT_PUTECHOFILT may be used prior to entering full duplex codec mode to set the echo canceller filter coefficients to a specified set of values.

The packet field PKT_GETECHOFILT may be used to retrieve the adapted filter coefficients after exiting codec mode.

The EC_FREEZE bit in ECONTROL flags, allows the echo canceller filter to be frozen. When the echo canceller is frozen, the filter is no longer adapted but echo cancellation can still occur. Enabling this bit can be used for test purposes. It could also be used as a means if saving power in cases where the echo canceller does not need to be continuously adapted.

The echo canceller is suitable for canceling the local echo caused by a 2-to-4 wire hybrid and can achieve echo cancellation of approximately 30 dB or more. Only the linear portion of the echo can be cancelled, so circuits should be designed to minimize nonlinearities. The Echo Return Loss (ERL) of the analog circuit must be 6dB or more for proper echo canceller operation. Linear Codecs will generally provide better performance than µ-law or a-law codecs due to lower quantization noise. The AMBE-4020™ Vocoder Chip employs an adaptive echo cancellation algorithm to cancel echoes of the decoder output present at the encoder input. The echo canceller is an adaptive LMS echo canceller with up to 16 milliseconds (128 samples) filter. It exceeds all the performance requirements specified by ITU-T recommendation G.165.

4.11 Echo Suppressor

In addition to an echo canceller, the AMBE-4020™ Vocoder Chip contains an echo suppressor. The echo suppressor can be enabled/disabled independently from the echo canceller using the ES_ENABLE bit in ECONTROL flags. See Table 40 ECONTROL Flags.

The echo suppressor may only be enabled for full duplex codec mode. Echo suppression is not supported in packet mode.

The echo suppressor can handle non-linearity that cannot be cancelled by the echo canceller. The echo suppressor consumes negligible power whereas the echo canceller consumes power proportional to its filter length. The maximum amount of suppression applied by the echo suppressor may be specified by sending the PKT_ECHOSUPLIM packet field prior to entering full duplex codec mode. By default, the suppression is limited to 30 dB. Note that the echo suppressor operates by attenuating the speech signal arriving at the encoder when a speech signal is detected at the decoder. This can be undesirable when speech is present in both directions simultaneously.

SECTION 5

5 Operating Modes

There are two primary modes, Packet Mode and Codec Mode, for the AMBE-4020™ vocoder chip. Packet Mode is the default mode after reset for boot configuration 0. Packet Mode is useful for configuring various parameters and can also be used for encoding speech packets to produce channel packets or decoding channel packets to produce speech packets. Codec Mode is used when it is desired to interface with the input speech signal from the ADC, DMIC, or I2S and/or an output speech signal on the DAC or I2S. Switching from Packet Mode to Codec Mode is achieved by sending a packet containing a PKT_STARTCODEC field. Switching from Codec Mode to Packet mode is achieved by sending a packet containing a PKT_STOPCODEC field or PKT_STOPCODECF field.

5.1 Packet Mode

In Packet Mode, there are three types of packets that can be sent to the AMBE-4020™ Vocoder Chip: control packets, speech packets and channel packets. When the AMBE-4020™ receives a control packet it typically sets a parameter or performs an action and then sends a response packet reply. Refer to Section 7.3.2 Control Packet Format, for details of all the control packets and their responses. When the AMBE-4020™ receives a speech packet, it is encoded using the currently selected vocoder rate, and a channel packet response is output. When the AMBE-4020™ receives a channel packet, it is decoded using the currently selected vocoder rate, and a speech packet is output in response.

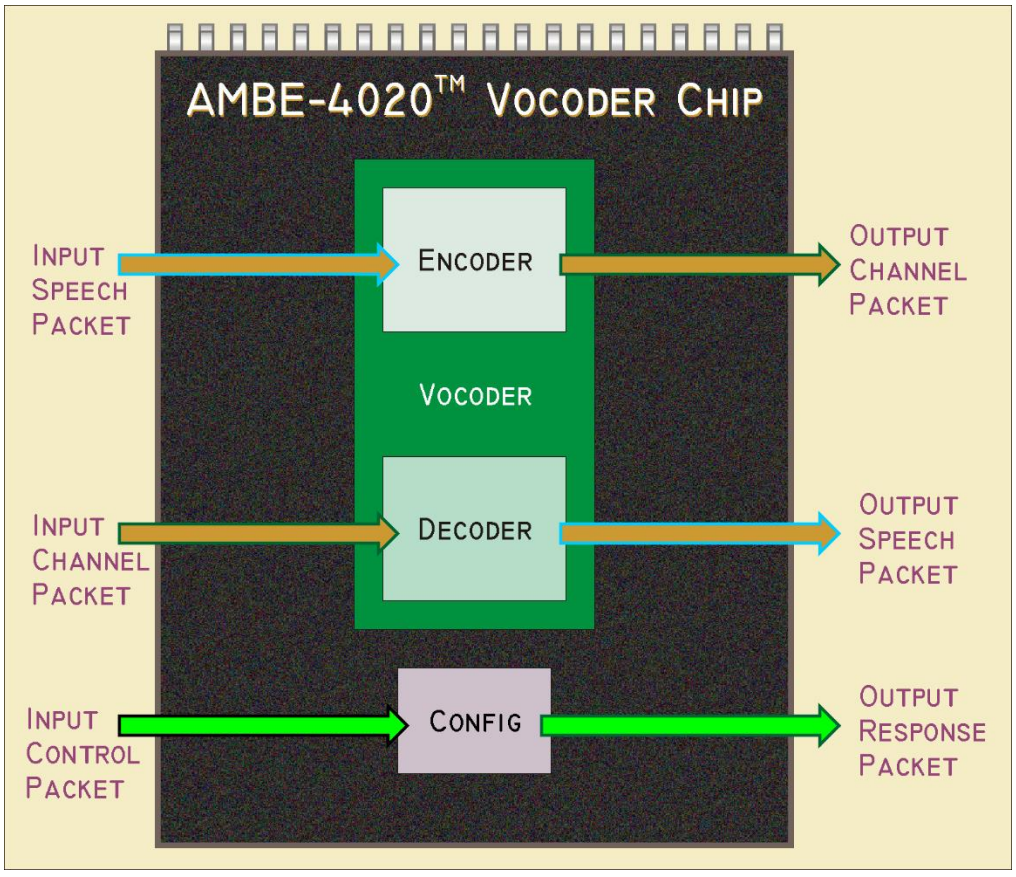


Figure 34 Packet Mode

Note that in packet mode both speech and channel data are transferred through the UART packet interface, whereas in codec mode the speech data is transferred through a separate interface such as ADC, DMIC, DAC, or I2S. It is the responsibility of

the designed system to extract the speech/channel data from these packets in order to pass the information to/from the codec/channel interfaces

The AMBE-4020™ contains a receive packet queue which can hold up to 1024 bytes. It also has a 1024-byte transmit packet queue (queue reduced to 512-byte for AMBE-4020™ Full-Duplex) where it places packet responses that are awaiting transmission. In Packet Mode, packets are processed in the order that they were received and the resulting response packets are transmitted in corresponding order. After the AMBE-4020™ Full-Duplex processes a packet; it places the response into the transmit queue and immediately begins processing the next packet in the receive queue. Typically, when sending control packets, it is best to send a single packet and then wait for its response packet before sending another packet. However, the AMBE-4020™ can continue to receive/transmit packets while it is still processing the prior packet. To increase throughput when processing speech and channel packets, it is often necessary to send the next input packet before the response from the prior packet has been received. For instance, to encode a series of speech packets: send the first two speech packets, and then continuously send another speech packet each time a channel packet is received. This method would allow speech packets to be encoded at a rate slightly faster than real-time, since the rate of packet input is directly tied to how quickly the output packet was produced. Another approach is to simply send speech packets at a fixed period of 20 ms, which results in speech packets being encoded in real-time. GFRAME may be useful for this purpose. A similar approach can be used for the decoder, except that channel packets are input to the AMBE-4020™ and speech packets are output.

When using the AMBE-4020™ Full-Duplex in packet mode for full-duplex communication, both speech packets and channel packets are input, resulting in both channel packets and speech packets are output. It is recommended that once per 20 ms, a speech packet followed by a channel packet is sent to the AMBE-4020™ Full-Duplex.

5.1.1 Flow Control in Packet Mode

In packet mode, the AMBE-4020™ does not utilize any packet timing signals. It processes incoming data packets as quickly as it can and sends back reply packets as soon as possible. After receiving a packet, in order to prevent the receive queue from overflowing, the AMBE-4020™ will set the UART_RTS pin high if less than 350 bytes of free space remain in the receive queue or if the total number of packets in the receive queue is equal to $(FLOWPKT)^8$. Setting the UART_RTS pin should succeed in preventing any overflows from occurring in the packet receive queue, however if it fails to prevent an overflow because the attached hardware ignored the UART_RTS signal, the AMBE-4020™ will discard the oldest (unprocessed) packets in the receive queue until there is sufficient space to store the newly received packet. When this occurs, entire packets are discarded, since partial packets cannot be interpreted by the AMBE-4020™.

If there is not enough space in the transmit queue to place a reply packet, then further processing of packets from the receive queue is stalled until sufficient space becomes available in the transmit queue. However, during such a stall in processing packets, the AMBE-4020™ can continue receiving packets until its receive packet queue fills.

Consider this scenario. The AMBE-4020™ is receiving a periodic stream of channel packets (one per 20 ms), when suddenly the UART_CTS signal is set high and is held there. For the case of this example assume that the transmit packet queue was empty at the time that UART_CTS was set high and that $(FLOWPKT)^8 = 3$ and that the size of the resulting speech packets are each 328 bytes. The AMBE-4020 will continue decoding the next 3 channel packets and will place the 3 resulting speech packets into the transmit queue. At that time the transmit queue will contain 984 bytes. The AMBE-4020 will continue to decode the next channel packet, but will need to stall further processing of packets until space becomes available in the transmit queue. However, it will continue to receive additional channel packets, until the receive queue becomes full, at which time the AMBE-4020™ will set its UART_RTS output indicating that it cannot accept any more packets. This example illustrates how flow control transfers from the transmit side of the AMBE-4020™ to the receive side of the AMBE-4020™ if flow is suddenly and completely stopped. After the AMBE-4020™ sets UART_CTS, this chain of flow stopping would presumably be transferred to the next device. Also note, that it resulted in both queues becoming filled (or nearly filled). More typically, the condition that caused flow stoppage would end before this occurred.

The following conditions may cause UART_RTS to be set (stopping flow):

1. After receiving a packet, the receive queue has less than 350 unused bytes, OR
2. After receiving a packet, the number of unprocessed packets is $\geq (FLOWPKT)^8$.

The following conditions may cause UART_RTS to clear (resuming flow):

1. After removing a packet from the receive queue, there is at least 350 unused bytes in the queue, AND

2. After removing a packet from the receive queue, the queue contains less than $(FLOWPKT)^8$ packets.

When UART_CTS is set, the AMBE-4020™ is prevented from transmitting packets. If UART_CTS is set while a byte is currently being transmitted, the transmission of the current byte will continue. Setting UART_CTS may cause a packet to be partially transmitted. After UART_CTS is cleared, the AMBE-4020™ will transmit the remainder of the packet.

5.1.2 Skew Control in Packet Mode

In packet mode, the number of speech samples in each speech packet is nominally 160. The number of samples in the input speech packet is specified within the PKT_SPEECHD packet field and can vary between 156 though 164 (inclusive). When sending channel packets to be decoded in packet mode, a PKT_SAMPLES fields can be used to specify the number of samples to be contained in the resulting speech packet. If a PKT_SAMPLES field is absent in the channel packet, the decoder will output 160 samples in the speech packet by default.

Typically the number of samples in a speech packet should be equal to 160 with an occasional adjustment to compensate for clock skew elsewhere in the system.

5.1.3 Packet Mode - Idle

When the AMBE-4020™ is in Packet Mode and has processed all received packets, such that there are no packets remaining in the packet queue it automatically enters Packet Mode - Idle to conserve power. Upon receiving a packet, it will automatically switch back to Packet Mode.

5.1.4 Low Power Packet Mode

The AMBE-4020™ enters Low Power Packet Mode from Packet Mode when a packet containing the PKT_PMODE field is received and $(PMODE)^8 = 3$. Prior to sending PKT_PMODE to enter this mode, the baud rate must be reduced to 125000 baud or lower. In this mode, the AMBE-4020™ power consumption is reduced by roughly 4x, but it can continue to process control packets, speech packets, and channel packets, but packet response time is reduced especially for speech packets and channel packets. It is not possible to run the encoder or decoder in real-time. The main purpose of this mode is to provide a way of reducing power during periods of packet inactivity. The AMBE-4020™ can be switched back to Packet Mode by sending a packet containing the PKT_PMODE field where $(PMODE)^8 < 3$. Note that you must not send PKT_BAUD field to change the baud rate while in Low Power Packet Mode. In addition, you may not send PKT_STARTCODEC to enter Codec Mode. It is necessary to switch to Packet Mode first.

5.1.5 Low Power Packet Mode - Idle

When the AMBE-4020™ is in Low Power Packet Mode and has processed all received packets, such that there are no packets remaining in the packet queue it automatically enters Low Power Packet Mode - Idle to conserve power. Upon receiving a packet, it will automatically switch back to Low Power Packet Mode.

5.1.6 Sleep Mode

The AMBE-4020™ enters Sleep Mode from Packet Mode when a packet containing the PKT_PMODE field is received and $(PMODE)^8 = 4$. The power savings is significant relative to Low Power Packet Mode. After entering Sleep Mode, no further packets may be sent to the AMBE-4020™ until a wake sequence takes place. To initiate waking the AMBE-4020™ from Sleep Mode, either transition the UART_CTS pin (low-to-high or high-to-low or pulse) or send a wake byte (any single character) via UART_RX. After initiating wakeup, wait until the AMBE-4020™ responds by sending a packet containing the PKT_READY field, after which time the AMBE-4020™ has re-entered Packet Mode and can resume normal operation. Upon waking from Sleep Mode, all prior configuration parameters are retained. Refer to Section 3.8: Wake Timing.

5.1.7 Halt Mode

The AMBE-4020™ enters Halt Mode from Packet Mode when a packet containing the PKT_PMODE field is received and $(PMODE)^8 = 5$. This is the lowest power mode offered by the AMBE-4020™. Power consumption is reduced slightly relative to Sleep Mode. One caveat is that the only way to exit this mode is via a hard reset (via RESET_n). While in Halt Mode, the

AMBE-4020™ does not respond to any packets. Resetting the AMBE-4020™, and possibly sending packets required to reconfigure it will consume significantly more time than waking from Sleep Mode. In Halt Mode, UART_RTS is held high to prevent incoming packet flow.

5.2 Codec Mode

Codec Mode has four variants: Encoder Codec Mode, Decoder Codec mode, Full Duplex Codec Mode (supported only by AMBE-4020™ Full Duplex) and push-to-talk codec mode. Encoder Codec Passthru mode and Decoder Codec Passthru Mode are also supported for test purposes.

5.2.1 Encoder Codec Mode

Entry into Encoder Codec Mode from Packet Mode is achieved by sending a packet containing the PKT_STARTCODEC field, with DUPLEX=DUPLEX_ENC. Upon entry into Encoder Codec Mode, the encoder begins acquiring 8 kHz samples from the selected speech interface (ADC, DMIC, or I2S) and outputs a channel packet once per 20 ms frame. The AMBE-4020™ will continue outputting channel packets every 20 ms until a packet containing a PKT_STOPCODEC field is received, at which time the AMBE-4020™ re-enters Packet Mode.

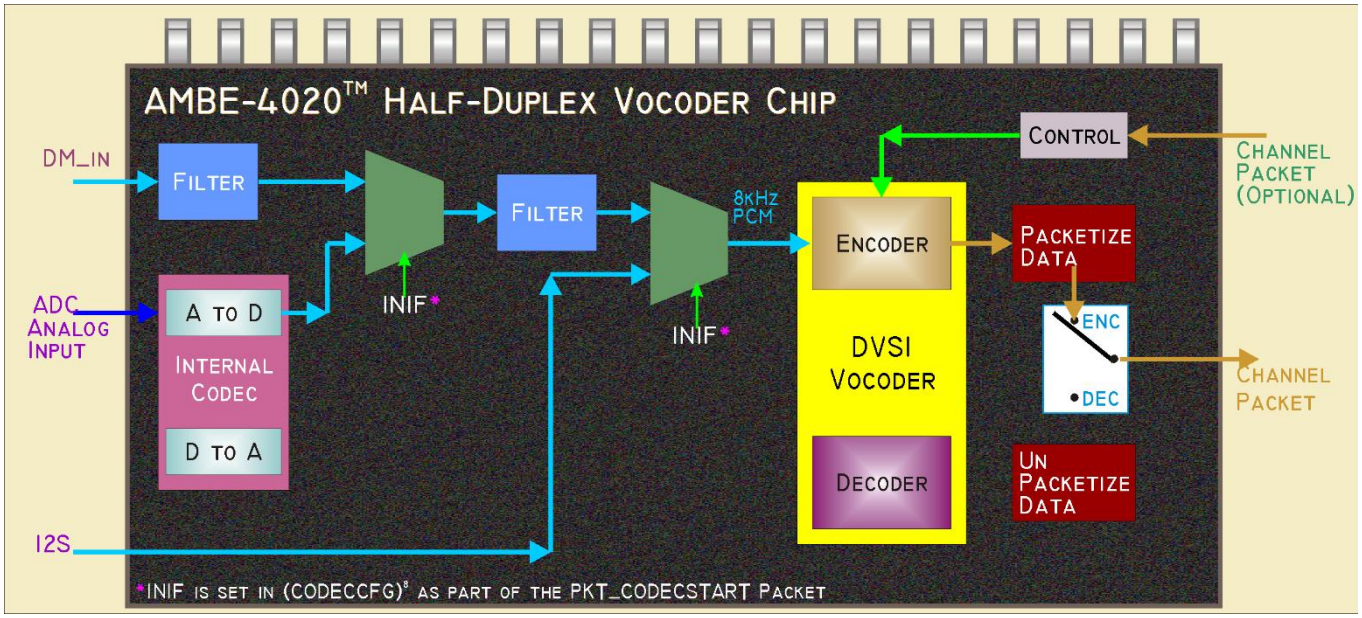


Figure 35 Encoder Codec Mode

5.2.2 Encoder Codec Passthru Mode

When the PASSTHRU bit in the PKT_STARTCODEC field is set, then the AMBE-4020™ enters a special passthru mode where it outputs a speech packet once per 20 ms rather than outputting channel packets containing compressed channel data. Each speech packet contains 160 (8 kHz) PCM samples. This feature may be useful for test purposes.

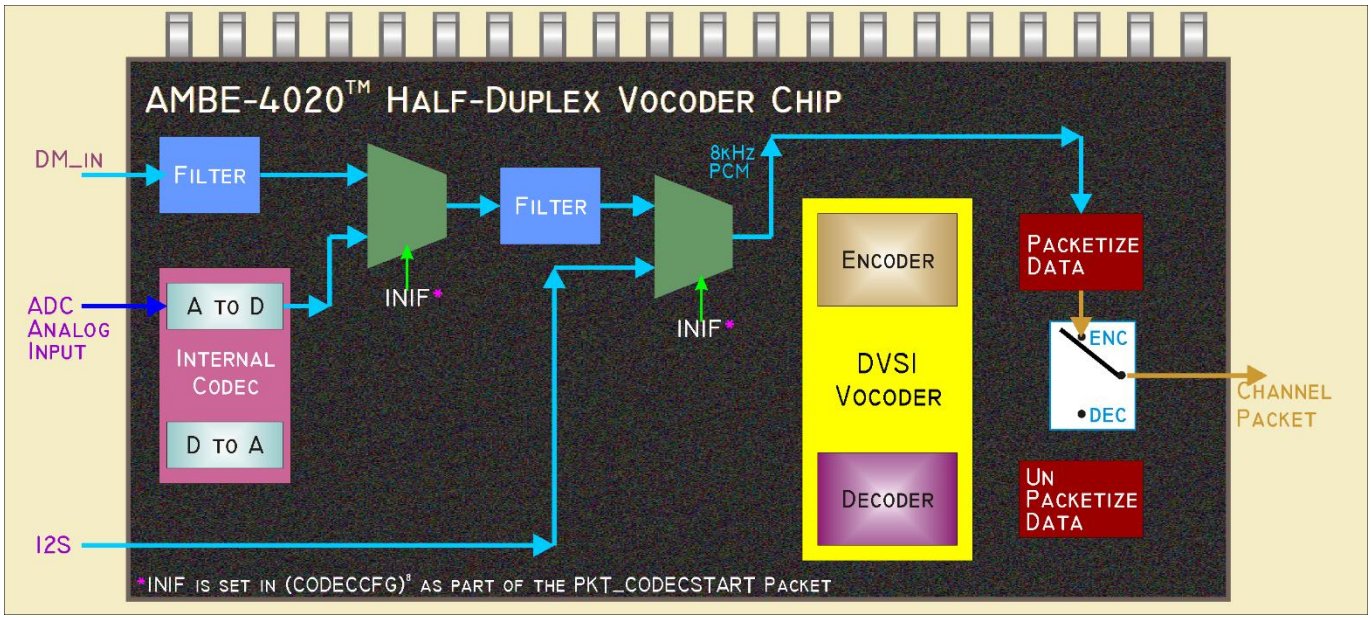


Figure 36 Encoder Pass thru mode

5.2.3 Decoder Codec Mode

Entry into Decoder Codec Mode from Packet Mode is achieved by sending a packet containing the PKT_STARTCODEC field, with DUPLEX=DUPLEX_DEC. Upon entry into decoder codec mode, the decoder expects to receive on channel packet every 20 ms. The decoder decodes the channel data contained in the channel packets and outputs the resulting 8 kHz speech data on the selected speech interface (DAC or I2S). The deadline for receiving a complete channel packet occurs at the falling edge of the OFRAME signal. If the decoder has no channel packets in its packet queue when the deadline is encountered then the decoder will automatically compensate for the lack of data by performing a frame repeat. If more than two consecutive decoder deadlines are encountered without receiving any channel data, then the decoder will insert comfort noise until channel packets are received. The parameter (NCHANPKT)[8] (default=1) specifies the maximum number of channel packets that the decoder will queue. If there are more channel packets than this in the queue when the decoder deadline is reached, then a frame of channel data is discarded. The decoder will continue consuming channel packets and outputting speech samples until a packet containing a PKT_STOPCODEC field is received, at which time the AMBE-4020™ re-enters Packet Mode.

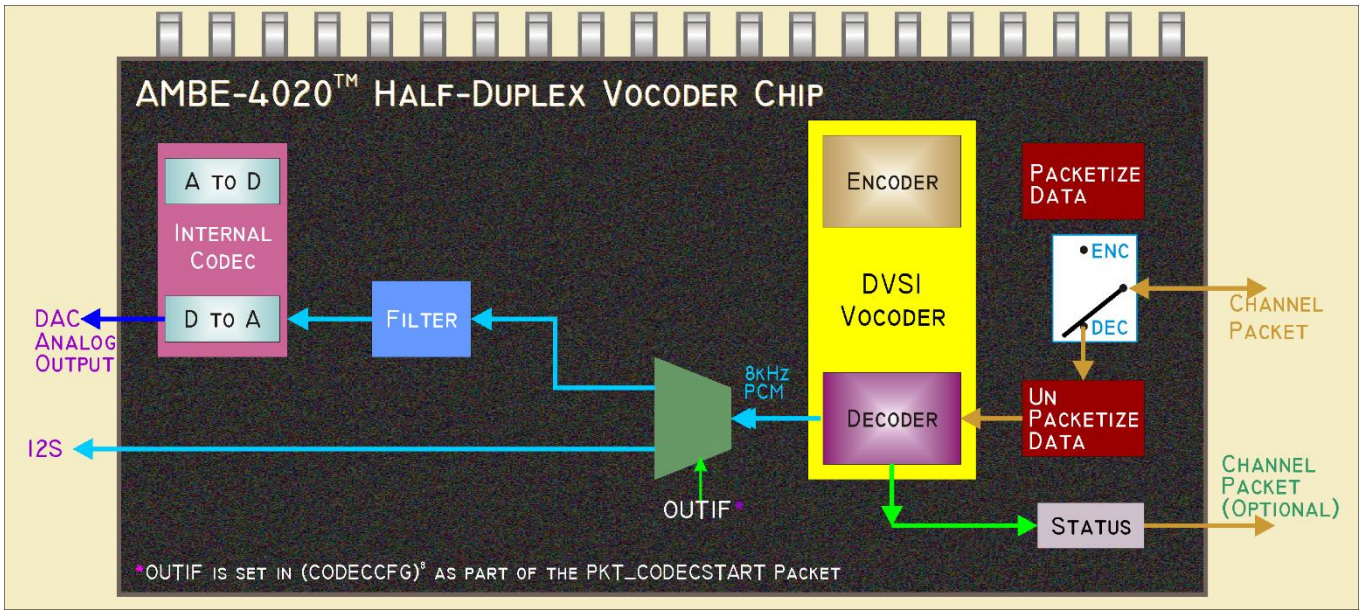


Figure 37 Decoder Codec Mode

5.2.4 Decoder Codec Passthru Mode

When the PASSTHRU bit in the PKT_STARTCODEC field is set, then the AMBE-4020™ enters a special passthru mode where it expects to receive a speech packet once per 20 ms rather than receiving channel packets containing compressed channel data. Each speech packet received must contain 160 (8 kHz) PCM samples. The speech samples are transferred out on the selected speech interface (DAC or I2S). This feature may be useful for test purposes.

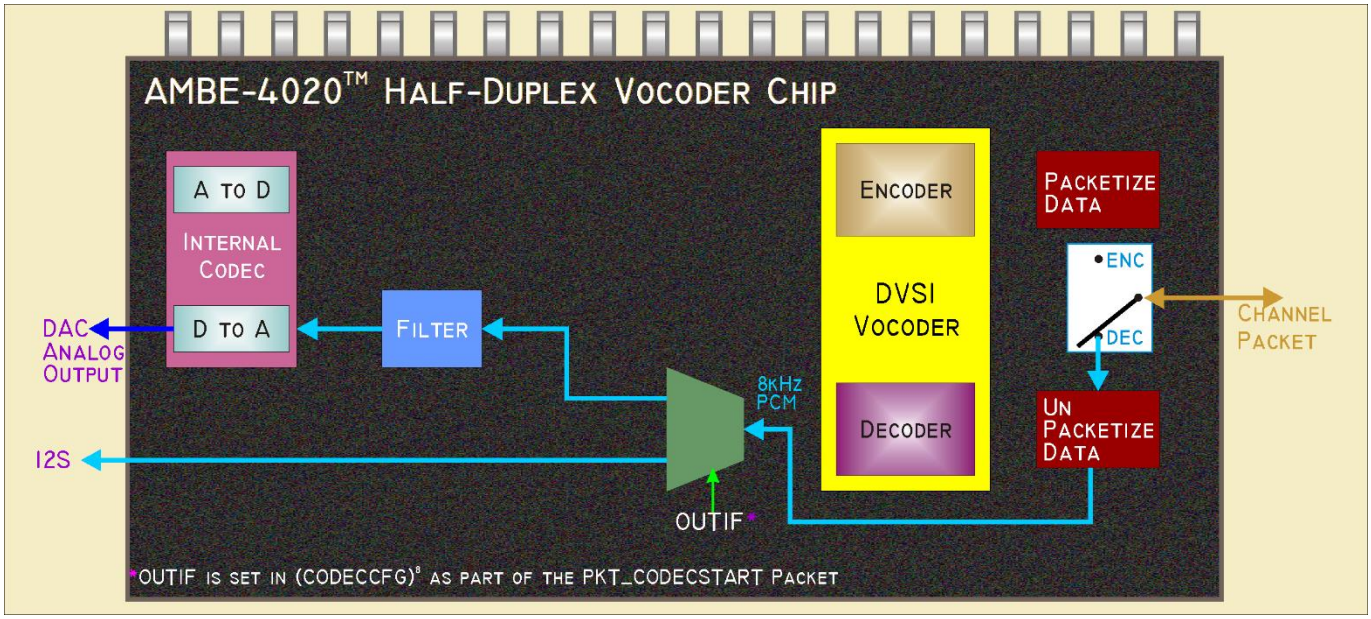


Figure 38 Decoder Passthru Mode

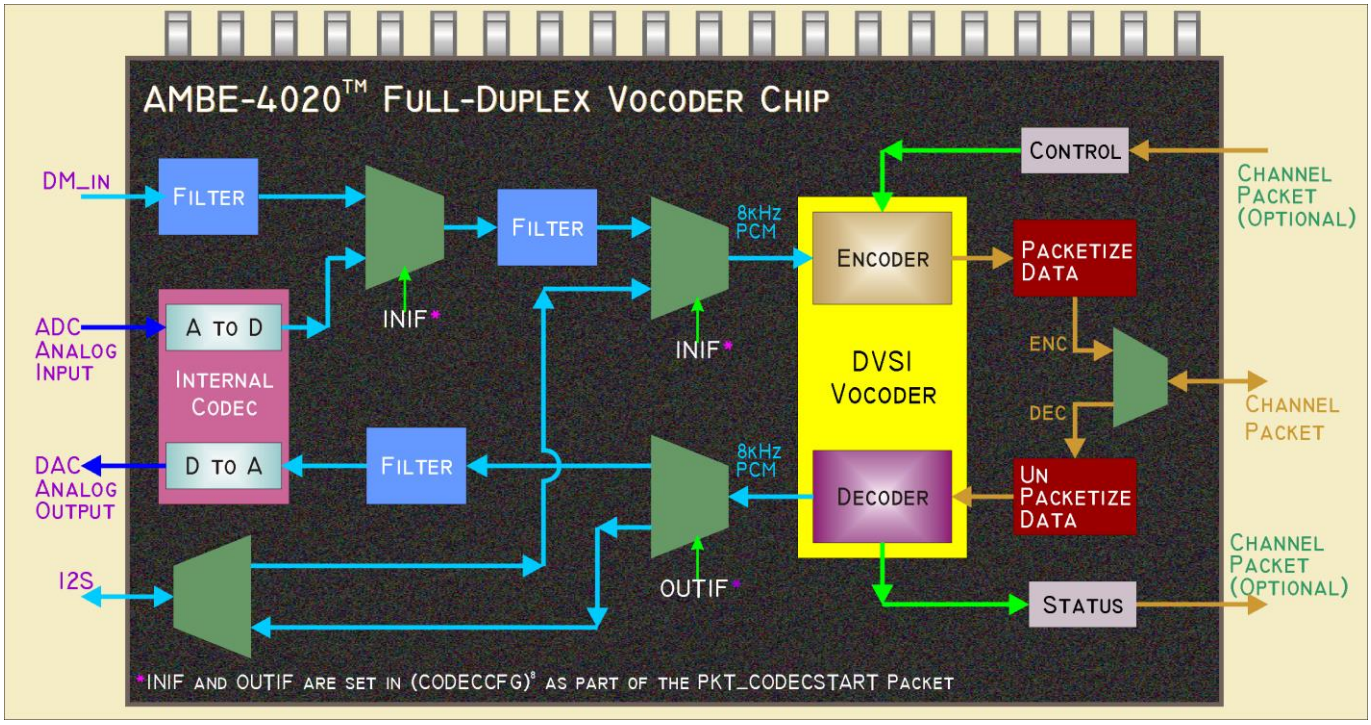


Figure 39 Full-Duplex Codec Mode

5.2.5 Full duplex Codec Mode

Entry into Full Duplex Codec Mode from Packet Mode is achieved by sending a packet containing the PKT_STARTCODEC field, with DUPLEX=DUPLEX_FULL.

Upon entry into Full Duplex Codec Mode, the encoder begins acquiring 8 kHz samples from the selected speech interface (ADC, DMIC, or I2S) and outputs a channel packet once per 20 ms frame.

The decoder is operational as well and expects to receive a channel packet every 20 ms. The decoder decodes the channel data contained in the channel packets and outputs the resulting 8 kHz speech data on the selected speech interface (DAC or I2S). The deadline for receiving a complete channel packet occurs at the falling edge of the OFRAME signal. If the decoder has no channel packets in its packet queue when the deadline is encountered then the decoder will automatically compensate for the lack of data by performing a frame repeat. If more than two consecutive decoder deadlines are encountered without receiving any channel data, then the decoder will insert comfort noise until channel packets are received. The parameter (NCHANPKT)[8] (default=1) specifies the maximum number of channel packets that the decoder will queue. If there are more channel packets than this in the queue when the decoder deadline is reached, then a frame of channel data is discarded

The encoder will continue outputting channel packets every 20 ms and the decoder will continue consuming channel packets until a packet containing a PKT_STOPCODEC field is received, at which time the AMBE-4020™ re-enters Packet Mode.

NOTE: The Full duplex codec supports only the following combinations of codec interfaces: I2S input with I2S output, ADC input with DAC output, or DMIC input with DAC output. For example, selecting I2S input with DAC output is invalid. However, using I2S input with DAC output is allowed for push-to-talk codec mode since the two interfaces are not used simultaneously

5.2.6 Push-to-Talk Codec Mode

Entry into Push-to-Talk Codec Mode from Packet Mode is achieved by sending a packet containing the PKT_STARTCODEC field, with DUPLEX=DUPLEX_PTT. Upon entry into this mode, the state of the ENC and DEC pins determine whether the encoder, decoder, or neither runs during any given period. When the encoder runs, it produces channel packets once per 20 ms in the same fashion as Encoder Codec Mode. When the decoder runs, it consumes packets and produces speech samples in the same fashion that Decoder Codec Mode does.

During periods where neither the encoder nor the decoder are running, the AMBE-4020™ conserves power by entering either PTT Codec Mode - Idle, PTT Codec Mode - Low Power, or PTT Codec Mode - Sleep, depending upon whether (PMODE)[8] is 0, 1, or 2. This mode is ideally suited to half-duplex systems, while at the same time providing a simple way of conserving power when neither the encoder nor decoder are required.

Note that it is not possible to select the encoder and decoder simultaneously. If ENC=DEC=1, then the encoder is selected. Table 38: ENC/DEC combinations for Push-to-Talk Codec Mode summarizes the various combinations of the ENC and DEC pins.

ENC LQFP Pin 44 BGA Pin G10	DEC LQFP Pin 45 BGA Pin G9	AMBE-4020™ State
0	0	Neither the encoder nor the decoder runs. The AMBE-4020™ enters the power saving mode specified by (PMODE)[8].
0	1	The decoder runs. The AMBE-4020™ should receive one channel packet every 20 ms.
1	0	The encoder runs. The AMBE-4020™ will produce one channel packet every 20 ms.
1	1	

Table 38: ENC/DEC combinations for Push-to-Talk Codec Mode

For more information regarding the ENC/DEC pins see Section 2.10 ENC/DEC Push-to-Talk Duplex Control Pins

5.2.7 Skew Control in Codec Mode

Skew control is enabled in Codec Mode by setting SKEW=1 within the PKT_STARTCODEC field used to enter Codec Mode. When skew control is enabled, two successive rising edges of the IFRAME signal (nominally 50 Hz) define the boundaries of a frame. The number of sample times within those boundaries is nominally 160. If the IFRAME rate is slightly greater than 50 Hz, then most frames will contain 160 samples, with an occasional frame containing 161 samples. If the IFRAME rate is slightly less than 50 Hz, then most frames will contain 160 samples, with an occasional frame containing 159 samples Moreover, if the IFRAME rate is constant in time, the period (in frames) between successive sample "slips" is predictable. Note that not only does IFRAME impact the number of samples per frame, it also controls the timing of the channel packets, since the encoder outputs one channel packet per frame or the decoder consumes one channel packet per frame. The IFRAME signal can be used by both the AMBE-4020™ and the transmitter, such that the AMBE-4020™ maintains sync with the transmitter and compensates for a lack of sync between the transmitter and the sample clocks.

In codec mode with skew control enabled, the 50 Hz IFRAME signal is used to delimit frame boundaries. The encoder receives samples at 8 Khz and the decoder produces samples at 8 kHz. The sample rate for I2S is ~8 Khz as determined from the CODEC_TX_FS/CODEC_RX_FS. The sample rates for ADC/DMIC/DAC are also 8 kHz where the 8 Khz sample rate is obtained by dividing down the 4 Mhz clock input. The IFRAME signal does not need to be synchronous with the sample clock. The IFRAME signal controls the rate at which the encoder produces channel packets and the rate at which the decoder consumes channel packets. The number of samples per 20 ms frame is automatically adjusted to account for differences between the sample clock and the IFRAME signal. The nominal number of samples per frame is 160 samples. This occurs in the ideal case where IFRAME is a perfect 50 Hz and the sample rate is exactly 8 kHz. However, the main purpose of skew control is to allow the non-ideal case.

Skew Control Scenario	Description
ideal case: IFRAME frequency = (sample rate)/160	The number of samples per frame is always 160.
IFRAME frequency is faster than sample rate IFRAME frequency > (sample rate)/160	The number of samples per frame is ≤ 160. The number of samples per frame is automatically determined by the AMBE4020 by checking how many samples have been clocked in/out between two successive rising edges of the IFRAME signal. In the close to ideal case where the IFRAME frequency is slightly greater than (sample rate)/160, most frames will have 160 samples, but an occasional frame will have only 159 samples.
IFRAME frequency is slower than sample rate IFRAME frequency < (sample rate)/160	The number of samples per frame is ≥ 160. The number of samples per frame is automatically determined by the AMBE4020 by checking how many samples have been clocked in/out between two successive rising edges of the IFRAME signal. In the close to ideal case where the IFRAME frequency is slightly less than (sample rate)/160, most frames will have 160 samples, but an occasional frame will have 161 samples.
IFRAME frequency drifts relative to the sample rate	The AMBE4020 can also handle the case where the relation between the IFRAME frequency and sample rate drift over time. There may be periods where IFRAME frequency > (sample rate)/160 and periods where IFRAME frequency < (sample rate)/160. The AMBE4020 accounts for this by simply counting the number of samples that occur between two rising edges of the IFRAME signal.

Table 39: Skew Control Scenarios

5.2.8 Timing of transmitted channel packets in Codec Mode

In Encoder Codec Mode, the AMBE-4020™ vocoder chip outputs one packet per 20 ms. The period of the OFRAME signal is 20 ms. The AMBE-4020™ begins transmitting a packet shortly after the falling edge of OFRAME. When skew control is disabled, the timing of the OFRAME is obtained by counting samples. When skew control is enabled, the rising edge of the OFRAME signal is synchronized to the rising edge of the IFRAME signal. The falling edge of the OFRAME signal occurs 80 (8 kHz) sample times after the rising edge of OFRAME.

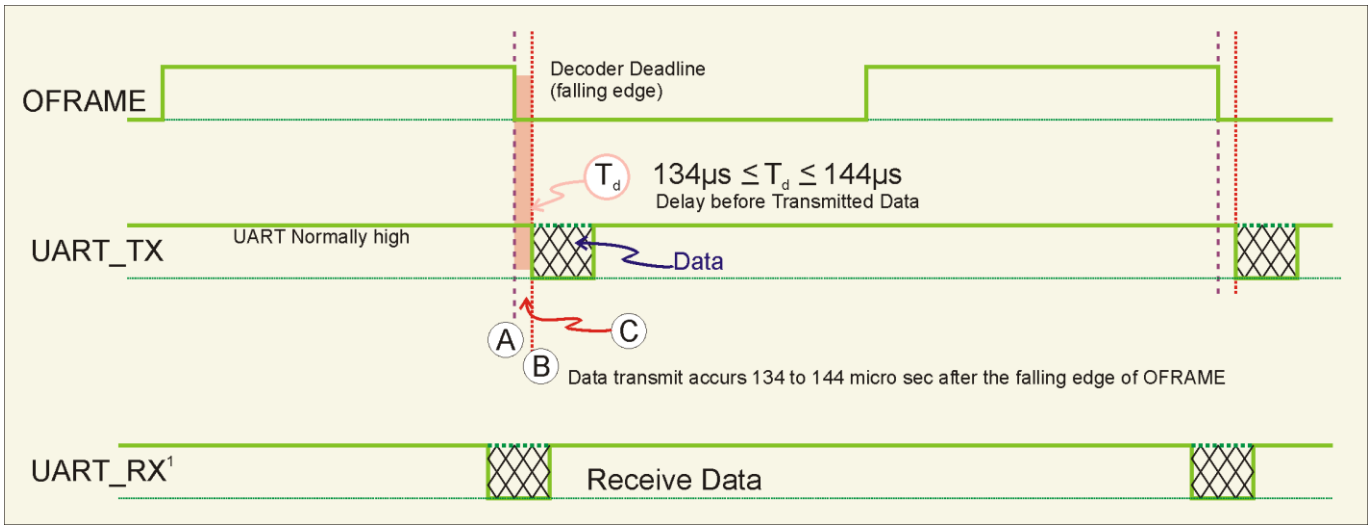


Figure 40 OFRAME Timing

Note: UART_RX is asynchronous to OFRAME, however, for best results packets should be received periodically (once per 20ms)

5.2.9 Timing of received channel packets in Codec Mode (Channel Packet Buffering)

As with Encoder Codec Mode, the OFRAME signal defines the frame boundaries. The decoder consumes a channel packet once per 20 ms at the falling edge of the OFRAME signal. The falling edge of OFRAME is therefore referred to as the decoder deadline. In order for a packet to be available for decoding, the entire packet must have been received prior to the decoder deadline.

A parameter named (NCHANPKT)[8] specifies the number of channel packets that are buffered by the decoder operating in Codec Mode. At every decoder deadline, the AMBE-4020™ checks the number of channel packets in the receive packet queue. If the number of packets exceeds (NCHANPKT)[8] then an extra channel packet is discarded (in addition to the one which will be consumed by the decoder) which results in a frame erasure. Discarding a packet is desired because it prevents the number of channel packets from growing unbounded over time. This would result in an accrued delay, since the decoder would be consuming frames at a rate slightly less than they are being received.

If the number of channel packets in the channel packet queue is zero, when the decoder deadline is reached then there is no packet for the decoder to consume. The decoder performs a frame repeat, which inserts a segment of speech derived from the latest set of speech model parameters maintained by the decoder.

Frame repeats and frame erasures are usually not noticeable. The frequency of erasures or repeats can be minimized by keeping the rate at which channel packets are presented to the AMBE-4020™ as close as possible to the rate at which it decodes them (see OFRAME).

The default for (NCHANPKT)[8] when boot configuration 0 is selected is 1. Increasing (NCHANPKT)[8] , effectively increases the amount of buffering and increases the worst case delay by 20 ms each time (NCHANPKT)[8] increases by 1. The minimum value for (NCHANPKT)[8] is 1, the maximum value for (NCHANPKT)[8] is 10. (NCHANPKT)[8] = 1, is suitable when packets are transferred to the AMBE-4020™ once per 20 ms, and there is very little jitter in the packet timing.

Increasing (NCHANPKT)[8] beyond 1 is practical in many situations. For situations where packets transferred to the AMBE-4020™ every 20 ms, but there is significant jitter in the packet timing (NCHANPKT)[8] = 2 would help. The delay increases by

20 ms, but when a packet arrives a little late, the AMBE-4020™ can fill the gap by using a frame from the buffer rather than employing a frame repeat.

Another situation that warrants increasing (NCHANPKT)[8] is a case where packets are received by the system in bursts. Possibly the system has a larger frame structure such that two or more 20 ms frames are produced by the demodulator at one time. Increasing (NCHANPKT)[8] in this case may not yield any net increase in the delay.

In a system where packets are received on average every 20 ms but there can be significant grouping of packets, then a large value for (NCHANPKT)[8] is useful, because the AMBE-4020™ can buffer up multiple packets such that it can handle larger delays. To decide on the value to choose for (NCHANPKT)[8] , consider that the optimal timing is to receive a packet once per 20 ms, then analyze the packet flow to determine the appropriate value for (NCHANPKT)[8].

5.3 Switching between Packet Mode and Codec Mode

A good technique for smooth operation and data transfer is to design the system so that the AMBE-4020™ Vocoder Chip boots into Packet Mode on start-up. Boot configuration 0, which is selected when BOOT1 = BOOT0 = 0, results in entry into Packet Mode after reset. This will allow the AMBE-4020™ Vocoder Chip to be sitting idle and ready to receive configuration packets. The user can then configure the AMBE-4020™ Vocoder Chip as needed. This method is beneficial because it puts the chip in a known state until it is ready to be utilized. Figure 41 Switching between Packet and Codec Modes shows a flow chart of the events needed to switch between the two modes.

Upon boot up after a reset the AMBE-4020™ Vocoder Chip is set to the mode (either codec mode or packet mode) as specified by the boot configuration data corresponding to the BOOT0/BOOT1pins. Switching the AMBE-4020™ Vocoder Chip from Packet Mode into Codec Mode or from Codec Mode into Packet Mode can be done by sending control packets.

To switch the AMBE-4020™ Vocoder Chip from Packet Mode into Codec Mode, a control packet with a PKT_STARTCODEC field (See Section Packet) must be sent to the AMBE-4020™ Vocoder Chip.

To switch the AMBE-4020™ Vocoder Chip from Codec Mode into Packet Mode, a control packet with a PKT_STOPCODEC field must be sent to the AMBE-4020™ Vocoder Chip

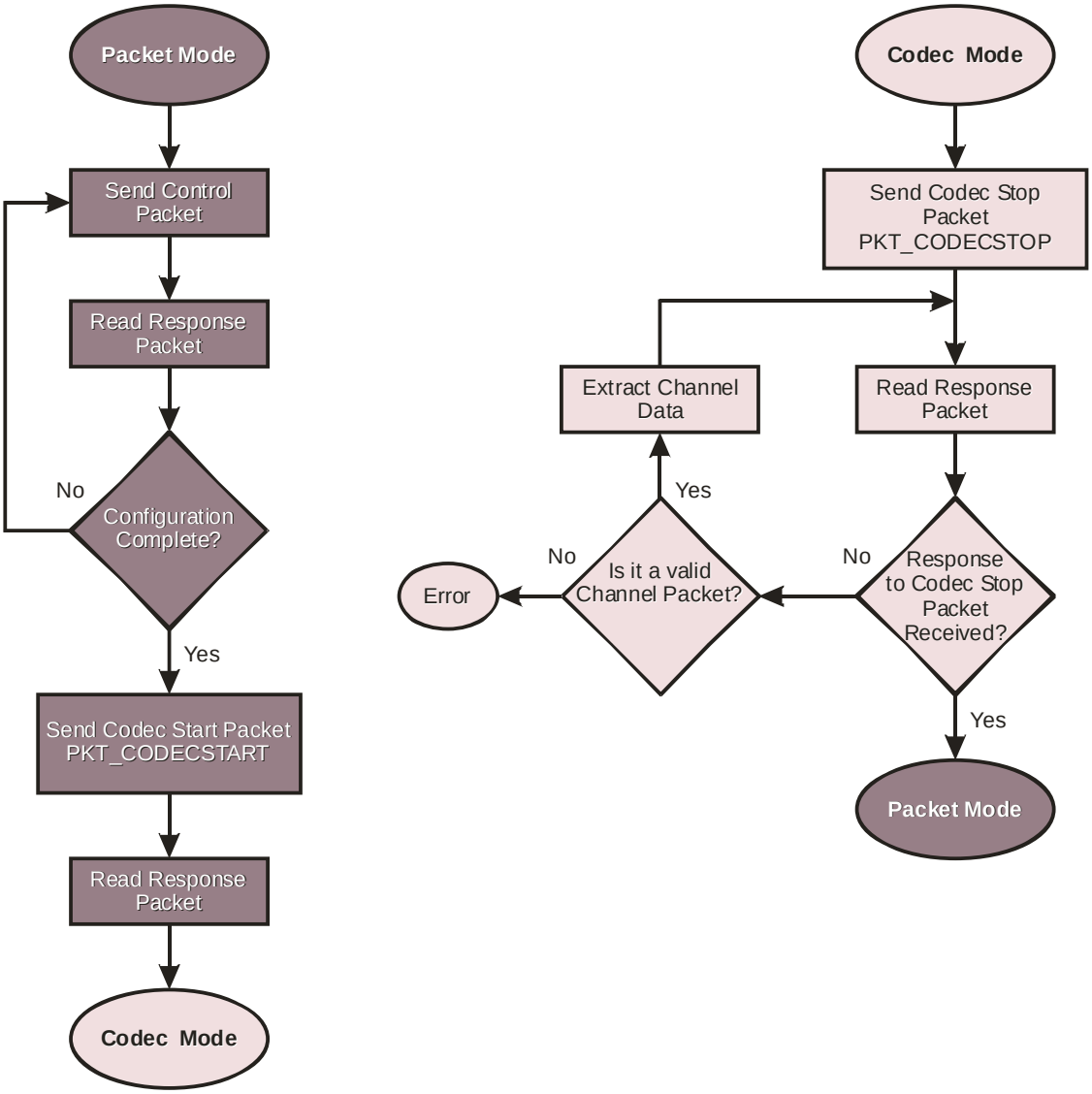


Figure 41 Switching between Packet and Codec Modes

SECTION 6

6 Vocoder Control/Status flags

The AMBE-4020™ passes 160±4 Codec samples to the encoder for each 20 ms frame. In addition to passing the speech samples to the encoder for every 20 ms frame, a 16-bit control word named ECONTROL is passed to the encoder.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
		EC_ENABLE		DTX_ENABLE		ES_ENABLE			NS_ENABLE		EC_FREEZE				

Table 40 ECONTROL Flags

Field	Description
15, 14, 13, 12, 10, 8, 7, 5, 3-0 Reserved	Reserved bits should be set to 0
13 EC_ENABLE	AMBE-4020™ Full-Duplex: Echo Canceller Enable 0 = Echo Canceller is disabled. 1 = Echo Canceller is enabled. AMBE-4020™ Half-Duplex: Reserved (set to 0)
11 DTX_ENABLE	Discontinuous Transmission Enable. 0 = the encoder does not output special silence frames when silence is detected. 1 = the encoder outputs a special silence frame whenever silence is detected.
9 ES_ENABLE	AMBE-4020™ Full-Duplex: Echo Suppressor Enable 0 = Echo suppressor is disabled. 1 = Echo suppressor is enabled. AMBE-4020™ Half-Duplex: Reserved (set to 0)
6 NS_ENABLE	Noise Suppressor Enable. 0 = Noise suppressor is disabled. 1 = Noise suppressor is enabled.
4 EC_FREEZE	AMBE-4020™ Full-Duplex: Echo Canceller Freeze. 0 = Do not freeze echo canceller filter. 1 = Freeze echo canceller filter. AMBE-4020™ Half-Duplex: Reserved (set to 0)

Table 41 ECONTROL Flags Description

ECONTROL is initialized at reset as determined by the selected boot configuration. It is also possible to directly specify the value for ECONTROL by sending a PKT_ECONTROL field within a configuration control packet prior entering codec mode. In addition, it is possible to specify ECONTROL every 20 ms by passing the value in every packet (or selected packets). Note that ECONTROL will retain its value until it is changed.

The encoder produces channel data for every 20 ms frame. The AMBE-4020™ places the channel data into an outgoing channel packet. The encoder also outputs a 16-bit status word named ESTATUS, for each 20 ms frame. The ESTATUS flags are specified in Table 43 ESTATUS Flags Description.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TONE_FRAME	Reserved				VAD	Reserved								VOICE_ACTIVE	Reserved

Table 42 ESTATUS Flags

Field	Description
15 TONE_FRAME	Indicates whether the encoder detected a single frequency tone, DTMF tone, KNOX tone, or call progress tone during the current 20 ms frame. 0 the encoder did not detect a tone in the current 20 ms frame. 1 the encoder detected a tone in the current 20 ms frame.
14-11 Reserved	These bits are reserved and the output value should be ignored.
10 VAD	
9-2 Reserved	These bits are reserved and the output value should be ignored.
1 VOICE_ACTIVE	If DTX is enabled, via the DTX_ENABLE bit of ECONTROL, then the encoder sets VOICE_ACTIVE=1 if the channel data for that frame must be transmitted. For frames which do not need to be transmitted, the encoder sets VOICE_ACTIVE=0. Note that when VOICE_ACTIVE=0, the encoder still produces a frame of channel data which may be transmitted if desired.
0 Reserved	This bit is reserved and its value should be ignored.

Table 43 ESTATUS Flags Description

By default, the ESTATUS flags are not output within the channel packets. If access to the flags is needed, it is possible to configure the AMBE-4020™ Vocoder Chip so that it will output the ESTATUS flags in every channel packet that is output or only when the ESTATUS flags change. The PKT_CHANFMT field within a configuration control packet is used to specify when/if the ESTATUS flags are output.

For each 20 ms frame, the AMBE-4020™ passes a 16-bit control word named DCONTROL to the decoder. DCONTROL is used to control various decoder features. Each bit of DCONTROL is summarized in DCONTROL is initialized at reset as determined by the selected boot configuration. It is also possible to directly specify the value for DCONTROL by sending a PKT_DCONTROL field within a configuration control packet prior to starting up the codec interface or running the decoder. In addition, it is possible to specify DCONTROL every 20 ms by passing the value in every packet (or selected packets). .Note that DCONTROL will retain its value until it is changed.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved												CNI_FRAME	LOST_FRAME	Reserved	

Table 44 DCONTROL Flags

Field	Description
15-4 Reserved	Reserved bits should be set to 0
3 CNI_FRAME	Comfort Noise Insertion Enable. 0 Comfort noise insertion is not requested for the current 20 ms frame. 1 Comfort noise insertion is requested for the current 20 ms frame. The decoder ignores any channel data provided to it and instead inserts comfort noise using the model parameters from the most recent valid silence frame that was received (or the default silence frame if no silence frames have been received yet).
2 LOST_FRAME	Frame Repeat Enable. 0 "Frame Repeat" is not requested.

	1 "Frame Repeat" is requested. The decoder ignores any channel data provided to it and instead uses the model parameters from the most recently received valid frame to synthesize a signal for the current frame.
1-0 Reserved	Reserved bits should be set to 0

Table 33 DCONTROL Flags Description

The AMBE-4020™ passes a frame of channel data, if available, to the decoder once every 20 ms. The decoder produces 160±4 speech samples for every 20 ms frame. In addition to outputting speech samples for each 20 ms frame, the decoder outputs a 16-bit status word named DSTATUS. The DSTATUS flags are as specified in Table 46 DSTATUS Flags. If AMBE-4020™ does not have a frame of channel data to pass to the decoder at the scheduled time, then the AMBE-4020™ forces the decoder to perform a frame repeat by setting the LOST_FRAME bit in DCONTROL for that frame only.

15	14|13|12|11|10|9|8|7|6	5	4 | 3 | 2	1	0
TONE_FRAME	Reserved	DATA_INVALID	Reserved	VOICE_ACTIVE	Reserved

Table 45 DSTATUS Flags

Field	Description
15 TONE_FRAME	Indicates when the decoder has received a tone frame. 0 The decoder did not receive a tone frame. 1 The decoder received a tone frame.
14-6 Reserved	These bits are reserved and the output value should be ignored.
5 DATA_INVALID	The decoder sets this bit whenever it performs a frame repeat. It also sets this bit if it inserted comfort noise due to channel errors or missing frames. The decoder will set DATA_INVALID=0 if it received a valid (voice, silence, or tone frame).
4-2 Reserved	These bits are reserved and the output value should be ignored.
1 VOICE_ACTIVE	The decoder sets VOICE_ACTIVE=1 if the decoder synthesized a voice frame or a tone frame. If the decoder synthesized a comfort noise frame, then it sets VOICE_ACTIVE=0. The decoder can synthesize comfort noise in the following circumstances: (a) A comfort noise frame (silence frame) was received by the decoder. (b) The decoder FEC (if enabled) found too many errors. (c) More than 2 consecutive frame repeats were requested.
0 Reserved	This bit is reserved and the output value should be ignored.

Table 46 DSTATUS Flags Description

SECTION 7

7 Packet Formats

7.1 Packet Interface Overview

The AMBE-4020™ can receive control packets, channel packets, and speech packets. It can output response packets, speech packets or channel packets.

The packet interface to the AMBE-4020™ Vocoder Chip is designed to provide as much flexibility as possible. The AMBE-4020™ Vocoder Chip always uses a packet format for the compressed voice data bits and for the chip configuration/control. All packets are transferred using the UART interface.

The AMBE-4020™ Vocoder Chip uses packets whether it is running in codec mode or packet mode. Prior to entering codec mode, packets are used for communicating with the AMBE-4020™ Vocoder Chip to configure the vocoder. After entering codec mode, packets are used to convey vocoder status/control information, as well as, transferring compressed voice bits from the encoder or to the decoder. When running in packet mode, control packets are still used to configure the vocoder and channel packets are still used to transfer channel data to/from the AMBE-4020™, but speech packets are also used to transfer speech data to/from the chip.

Every packet includes a HEADER that consists of a START byte for identification of the beginning of the packet, LENGTH data to indicate how many bytes are in the packet and a TYPE byte that specifies what to do with the packet. Packets are processed in a first-in-first-out manner.

7.2 Packet Interface Format

The AMBE-4020™ Vocoder Chip's UART is its only interface that handles packets. The AMBE-4020™ Vocoder Chip supports packets with a parity field or packets without a parity field. The packet format is as shown in Table 47 General Packet Format. A packet always starts with a PACKET HEADER byte. The next two bytes contain the PACKET LENGTH and the next byte contains the PACKET TYPE. Each packet can contain one or more fields that are shown as $FIELD_0$ through $FIELD_{N-1}$ in Table 47. By default, parity fields are enabled after reset. When parity is enabled, the final field ($FIELD_{N-1}$) must be a parity field (See **PKT_PARITYBYTE** .)

General Packet Format [1]					
Packet Header			Fields		
START_BYTE	**LENGTH**	**TYPE**	**$FIELD_0$**	**…**	**$FIELD_{N-1}$**
1 byte	2 bytes	1 byte	L_0 bytes	…	L_{N-1} bytes
0x61	LLLL	TT			
[1] Note that when parity is enabled, the final field ($FIELD_{N-1}$) must be a 2-byte parity field					

Table 47 General Packet Format

7.2.1 START_BYTE (1 byte)

Referring to Table 47 General Packet Format, the START_BYTE byte always has a fixed value of 0x61.

7.2.2 LENGTH (2 bytes)

Referring to Table 47 General Packet Format, the PACKET LENGTH occupies the second two bytes of the packet. The MS byte of the packet length is the second byte of the packet and the LS byte of the packet length is the third byte of the packet. To calculate the PACKET LENGTH take the sum of L_0 through L_{N-1}. Do not include the 4 bytes (START_BYTE, PACKET LENGTH, and PACKET TYPE) from the Packet Header in the PACKET LENGTH. Therefore, in Table 47 General Packet Format, the PACKET LENGTH is the sum of L_0 through L_{N-1}. Note that the PACKET LENGTH excludes the first 4 bytes taken up by the START_BYTE, PACKET LENGTH, and PACKET TYPE. PACKET LENGTH is therefore the total length (in bytes) of the entire packet minus 4 bytes.

7.2.3 TYPE (1 byte)

Referring to Table 47 General Packet Format, the PACKET TYPE occupies the fourth byte of every packet.

There are 3 different packet types for the AMBE-4020™ vocoder chip.

Packet Types		
Packet Name	Description	Type Value (Hex)
Control Packet	Used to setup chip modes, rates, configure hardware, initialize encoder/decoder, enable low-power mode, specify output packet formats, etc. When a control packet is received, the chip returns a control packet with response fields that contain response data for some control packets or indication of errors in the control packet.	0x00
Speech Packet	In Packet Mode, these packets are used to input speech data to the encoder and to output speech data from the decoder. In addition to speech data, the packet can provide flags to control the encoder operation on a frame-by-frame basis. The speech packet also can have a field that forces the encoder to produce a tone frame.	0x02
Channel Packet	These packets are used to input channel data to the decoder and to output channel data from the encoder. In addition to channel data, the packet can provide flags that control the decoder operation on a frame-by-frame basis. A channel packet can also contain a field that forces the decoder to produce a tone frame.	0x01

Table 48 Packet Types

7.2.4 Packet Fields

Referring to Table 47 General Packet Format, the remainder of a packet after the START_BYTE, LENGTH, and TYPE is made up of packet fields. The packet fields contain the useful packet information. Various different packet fields each with their own format are defined in the next sections, however, the general format of a field is shown in Table 49 General Field Format.

A field consists of a field identifier followed by field data. The length of field data is dependent upon the field identifier. Many fields have fixed lengths. Some fields, such as those that contain speech samples or channel data are variable in length; and in such cases, the length of the field data is embedded inside field data.

Field - Packet Format	
Field Identifier	Field Data
1 byte	L_n-1 bytes

Table 49 General Field Format

7.2.4.1 Parity Field (Parity is enabled by default)

A Parity Field is a special 2-byte field. When used it must be the last field of the packet. When parity fields are enabled, the AMBE-4020™ Vocoder Chip inserts a 2-byte field at the end of all output packets. The first byte of the parity field is the parity field identifier and is always equal to 0x2f. The second byte of the parity field is the parity byte. It is obtained by "Exclusive-oring" every byte in the packet, except for the START_BYTE and the PARITY_BYTE, together. If parity fields are enabled, the AMBE-4020™ Vocoder Chip checks the parity byte for all received packets and discards any packet that has an incorrect parity byte. Parity fields can be enabled or disabled (for all future input and output packets) by sending a PKT_PARITYMODE field in a control packet.

7.3 Packet Field Reference

7.3.1 Packet Field Nomenclature Key

The description of the packet fields uses a quick reference format to easily see the details of the packet and what data it contains.

Packet Fields Description Example:

{ PKT_NAME = 0xNN, $(DAT_0)^{LEN0}$ …$(DAT_n)^{LENn}$} =>{ PKT_NAME, RESP}

PKT_NAME
The PKT_NAME is the field identifier name as shown in Table 50 Packet Fields

0xNN (Field_ID)
The Field_ID is the Field Identifier code. It is in Hex format and shown as 0x00.

DAT_0 ……. DAT_n (can be more than one data value)
The DAT_n is the value of the control field for the PKT_NAME. Some packets may have no data value field, while others may have one or more.

LEN_n
The LEN_n superscript is the number of bits for the given Data Value.

Returns (=>)
The => Indicates the following information is the expected Response PKT from the PKT type sent

PKT_NAME
The name of the Response Packet.

RESP (Returned Value)
This is the value of the data in the Response packet.

Packet Fields										
Field Identifier Name	Field Identifier Code	Field Length (bytes)	Response Field Length (bytes)	Supported by Packet Types: Control Input	Control Response	Channel Input	Channel Output	Speech Input	Speech Output	Description
PKT_COMPAND	0x32	2	2	☑	☑					Companding ON/OFF and a-law/μ-law selection
PKT_RATET	0x09	2	2	☑	☑					Select rate from table.
PKT_RATEP	0x0A	13	2	☑	☑					Select custom rate.
PKT_INIT	0x0B	2	2	☑	☑					Initialize encoder and/or decoder.
PKT_PMODE	0x10	2	2	☑	☑					Enter a different power mode or specify power saving mode to be used during push-to-talk codec mode.
PKT_I2CDATA	0x38	V[1]	2	☑	☑					Specifies configuration data for codec to be sent via I2C (but does not actually send the I2C data).
PKT_STARTCODEC	0x2A	2	2	☑	☑					Switches from packet mode to codec mode
PKT_STOPCODEC	0x2B	1	2	☑	☑					Switches from codec mode to packet mode
PKT_STOPCODECF	0x2C	1	2	☑	☑					Switches from codec mode to packet mode, but flushes out all channel packets before sending a response packet.
PKT_CHANFMT	0x15	5	2	☑	☑					Sets the format of the output channel packet
PKT_SPCHFMT	0x16	5	2	☑	☑					Sets the format of the output speech packet
PKT_PRODID	0x30	1	V[1]	☑	☑					Query for product identification
PKT_VERSTRING	0x31	1	V[1]	☑	☑					Query for product version string
PKT_READY	0x39	NA[2]	1		☑					Indicates that the device is ready to receive packets
PKT_ERROR	0xFF	NA[2]	2		☑					Indicates that an error condition occurred.
PKT_RESET	0x33	1	NA[2]	☑						Reset the device using hard configuration via pins.
PKT_PARITYMODE	0x3F	2	2	☑	☑					Enable (default) / disable parity fields
PKT_WRITE I2C	0x44	V[1]	2	☑	☑					Writes to an I^2C device such as a codec
PKT_READI2C	0x45	V[1]	V[1]	☑	☑					Reads from an I^2C device such as a codec
PKT_CONFIGI2C	0x43	V[1]	2	☑	☑					Configures I^2C settings of the AMBE-4020™
PKT_CLRCODECRESET	0x46	1	2	☑	☑					Sets the codec reset signal to Low
PKT_SETCODECRESET	0x47	1	2	☑	☑					Sets the codec reset signal to High
PKT_DISCARDCODEC	0x48	3	2	☑	☑					Number of codec samples to discard

PKT_DELAYNUS	0x49	3	2	☑	☑					Delays the next control field processing (in microseconds)
PKT_ERRTHRESH	0x1A	5	2	☑	☑					Specify custom error mitigation thresholds
PKT_BOOTCFG	0x0C	130	2	☑	☑					Sets the EEPROM boot configuration
PKT_BOOTCFG	0x0C	2	129	☑	☑					Reads EEPROM boot configuration
PKT_BAUD	0x57	4	2	☑	☑					Changes the baud rate of the UART interface
PKT_GPIO	0x5F	3	2	☑	☑					For GPIO on pins GPIO0, GPIO1, GPIO2, or GPIO3
PKT_UFRAME	0x5E	3	2	☑	☑					Allows the rising edge and falling edge of the UFRAME pin to be specified relative to OFRAME.
PKT_BREAKF	0x60	2	2	☑	☑					Allows the function of the UART break signal to be specified
PKT_NCHANPKT	0x5B	2	2	☑	☑					Configures the number of received channel packets that are buffered during codec mode
PKT_GDIV	0x5C	3	2	☑	☑					Enables generation of the GFRAME signal and specifies the frequency of the signal
PKT_FLOWPKT	0x4E	2	2	☑	☑					Maximum number of packets to be received before UART_RTS is set high
PKT_ECHOLEN	0x64	3	2	☑	☑					Sets the length of the echo canceller filter
PKT_PUTECHOFILT	0x65	V[1]	2	☑	☑					Sets the length of the echo canceller filter and specifies filter coefficients
PKT_GETECHOFILT	0x66	1	V[1]	☑	☑					Retrieves the current echo canceller filter coefficients.
PKT_ECHOSUPLIM	0x69	2	2	☑	☑					Specifies maximum attenuation by the echo suppressor.
PKT_PARITYBYTE	0x2F	2	2	☑	☑	☑	☑	☑	☑	Specifies that the following byte is a parity byte.
PKT_ECONTROL	0x05	3	2	☑	☑	☑		☑		Encoder control flags.
PKT_DCONTROL	0x06	3	2	☑	☑	☑		☑		Decoder control flags.
PKT_CONTROL	0x5D	4	2	☑	☑	☑		☑		Additional way to alter either ECONTROL and DCONTROL
PKT_GAIN	0x4B	3	2	☑	☑	☑				Used to set Input gain and output gain to be anywhere between +20 and -20 dB
PKT_ESTATUS	0x05	NA[2]	3				☑			Reports status flags from encoder
PKT_DSTATUS	0x06	NA[2]	3				☑		☑	Reports status flags from decoder
PKT_TONERCV	0x53	NA[2]	3				☑		☑	Reports frequency and amplitude of tone received by the decoder.
PKT_BER	0x56	NA[2]	7				☑		☑	Reports error mitigation data from the FEC decoder.
PKT_TONEDET	0x52	NA[2]	3				☑			Reports frequency and amplitude of tone detected by the encoder.
PKT_ECHOFILT	0x67	NA[2]	V[1]				☑			Reports the echo canceller filter coefficients from the echo canceller.

PKT_ECHOSTAT	0x68	NA[2]	7				☑			Reports echo canceller attenuation, ERL, and signal level.
PKT_LVL	0x23	NA[2]	9				☑			Reports signal levels received by the encoder and output by the decoder.
PKT_PKT	0x24	NA[2]	29				☑			Reports various packet statistics.
PKT_SAMPLES	0x03	2	2			☑	☑			1. Specifies the number of samples that the decoder should output in packet mode. 2. Reports the number of samples per frame in codec mode.
PKT_TONEXMT	0x50	4	NA[2]			☑		☑		Specifies frequency, amplitude, and duration of tone to be transmitted by the encoder.
PKT_TONEGEN	0x51	4	NA[2]			☑				Specifies frequency, amplitude and duration of tone to be generated by the decoder.
PKT_CHAND	0x01	V[1]	NA[2]			☑	☑			Contains compressed channel data output by the encoder or input to the decoder.
PKT_CHAND4	0x17	V[1]	NA[2]			☑				Contains compressed channel data for input to the decoder in four-bit soft-decision format.
PKT_SPEECHD	0x00	V[1]	V[1]					☑	☑	In packet mode, contains speech for input to the encoder or speech output by the decoder. In encoder codec passthru mode contains speech obtained from codec input interface (I2S / ADC / DMIC). In decoder codec passthru mode samples from the packet are passed directly to the specified codec output interface (I2S/DAC).

NOTES:
[1] Varies. The length of the field varies. See documentation for the field for more information.
[2] Not Applicable. The length of the field is not applicable, because it is not supported in this direction.
Control + Click to follow Hyperlink
Alt + left arrow to return back to the original location.

Table 50 Packet Fields

PKT_COMPAND field Enables/Disables the use of companded data and allows for selection or either a-law or μ-law companding. Companding may be enabled when using packet mode or when using codec mode with the I2S interface. When enabled, it applies to both the speech input and the speech output. It is not possible to specify companding for the input and output independently.

{ PKT_COMPAND = 0x32, (COMPAND)[8] } => { PKT_COMPAND, 0x00 }

This field is only supported by control packets.

Options for PKT_COMPAND Field	
Description	**$(COMPAND)^8$ Value**
Select μ-law companding	2
Select a-law companding	3
Companding Disabled	0

Table 51 PKT_COMPAND Field Options

PKT_RATET field specifies one of the built-in rates. Sets a built-in Rate from Table 71 Rate Index Numbers

{ PKT_RATET = 0x09, $(RATET)^8$ } => { PKT_RATET, 0x00 }

The coding rate can be modified for both the encoder and the decoder by sending a PKT_RATET or PKT_RATEP packet. Table 71 Rate Index Numbers shows standard Rate / FEC combinations

Table 71 Rate Index Numbers lists the predefined values for various source and FEC rates that are built into the AMBE-4020™ Vocoder Chip. The table also indicates what rates are compatible with older DVSI vocoder chips such as the AMBE-2000™ Vocoder Chip (using AMBE™+ technology) and the AMBE-1000™ Vocoder Chip (using AMBE™ technology).

This field is only supported by control packets.

PKT_RATEP field specifies Custom Rate words. If rates other than those indicated in Table 71 Rate Index Numbers and Settings are desired then the PKT_RATEP field must be used to specify a custom rate. Custom rate words can be obtained by contacting DVSI with your system requirements.

{ PKT_RATEP = 0x0A, $(RCW_0)^{16}$, $(RCW_1)^{16}$, $(RCW_2)^{16}$, $(RCW_3)^{16}$, $(RCW_4)^{16}$, $(RCW_5)^{16}$ } => {PKT_RATEP, 0x00 }

Example of a PKT_RATEP field with the custom rate of 2800 bps voice and 0 bps FEC:

{ 0x0A, 0x0038, 0x0765, 0x0000, 0x0000, 0x0000, 0x0038 }

This field is only supported by control packets.

PKT_INIT field initializes the encoder and/or decoder depending upon the value of $(INIT)^8$.

{ PKT_INIT = 0x0B, $(INIT)^8$ } => { PKT_INIT, 0x00 }

$(INIT)^8$ is comprised of bits as shown in Table 52 PKT_INIT Field.

7	6	5	4	3	2	1	0
Reserved					INIT_EC	INIT_DEC	INIT_ENC

Field	Description
7, 6, 5, 4. 3	Reserved

2 INIT_EC	Echo Canceller Initialization 0 = Echo Canceller not Initialized 1 = Echo Canceller Initialized
1 INIT_DEC	Decoder Initialization 0 = Decoder not Initialized 1 = Decoder Initialized
0 INIT_ENC	Encoder Initialization 0 = Encoder not Initialized 1 = Encoder Initialized

Table 52 PKT_INIT Field

This field is only supported by control packets.

PKT_PMODE field tells the AMBE-4020™ Vocoder Chip to enter a different power mode upon receipt of the packet or tells the chip how to save power during PTT codec mode.

{ PKT_PMODE = 0x10, $(PMODE)^8$ } => { PKT_PMODE, 0x00 }

Setting $(PMODE)^8$ to 0x03, 0x04 or 0x05 has the immediate effect of entry into a lower power mode. Setting $(PMODE)^8$ to either 0x00, 0x01 or 0x02 does not have an immediate impact on power. It merely effects the power savings after the chip is subsequently switched into push-to-talk codec mode. There is no difference between $(PMODE)^8$ = 0, 1, 2 unless push-to-talk codec mode is used.

This field is only supported by control packets.

Options for PKT_PMODE Field	
Description	**$(PMODE)^8$ Value**
Default Power Mode	0x00
VLPW while PTT is idle	0x01
LLS while PTT is idle	0x02
entry into VLPR/VLPW	0x03
entry into LLS	0x04
entry into VLLS	0x05

Table 53 PKT_PMODE Field Settings

PKT_I2CDATA field contains configuration data that the AMBE-4020™ will send to the codec via I^2C after it receives a PKT_STARTCODEC packet. $(NR)^8$ (where $0 < (NR)^8 < 10$) specifies the number of registers that will be specified by the remainder of the packet field. $(NR)^8$ must be followed by exactly $(NR)^8$ [$(R_n)^8$, $(D_n)^8$] pairs. $(R_n)^8$ specifies the register number to be set and $(D_n)^8$ specifies the value for the register. The I^2C configuration data specified via this field is sent to initialize the codec after {PKT_STARTCODEC} field is received provided that the I^2S codec interface is selected.

{ PKT_I2CDATA = 0x38, $(NR)^8$ [, $(R_0)^8$, $(D_0)^8$] ... [, $(R_{NR-1})^8$, $(D_{NR-1})^8$] } => { PKT_I2CDATA, 0x00 }

Example 1:

The field { 0x38, 0x05, 0x01, 0x41, 0x02, 0xA0, 0x04, 0x83, 0x05, 0xb8, 0x06, 0x20 } specifies that 5 registers are configured via I^2C. Register 1 is set to 0x41, register 2 is set to 0xA0, register 4 is set to 0x83, register 5 is set to 0xB8, and register 6 is set to 0x20.

Example 2:

The field { 0x38, 0x00 } specifies that no registers are to be initialized via I^2C after {PKT_STARTCODEC,...} is received.

This field is only supported by control packets.

PKT_STARTCODEC field will switch the AMBE-4020™ Vocoder Chip from packet mode to codec mode.

{ PKT_STARTCODEC = 0x2A, $(CODECCFG)^8$ } => { PKT_STARTCODEC, 0x00 }

$(CODECCFG)^8$ is comprised of a 2-bit duplex field, a 2-bit OUTIF field, a 2-bit INIF field, a 1-bit passthru field, and a 1-bit skew field as shown in table TBD.

7	6	5	4	3	2	1	0
DUPLEX		OUTIF		INIF		PASSTHRU	SKEW

Field	Description
7-6 DUPLEX	select encoder/decoder or full duplex 00 = DUPLEX_PTT half-duplex as determined by ENC/DEC pins 01 = DUPLEX_DEC half-duplex decoder 10 = DUPLEX_ENC half-duplex encoder 11 = RESERVED for AMBE-4020™ 11 = DUPLEX_FULL Full duplex operation on the AMBE-4020™ Full-Duplex
5-4 OUTIF	select interface for speech output 00 = OUTIF_I2S speech is output via I2S interface 01 = OUTIF_DAC speech is output on the DAC pin 10 = RESERVED 11 = RESERVED
3-2 INIF	select interface for speech input 00 = INIF_I2S speech is input via I2S interface 01 = INIF_ADC speech is input from ADC interface 10 = INIF_DMIC speech is input from the DMIC interface 11 = RESERVED
1 PASSTHRU	select whether passthru is enabled 0 = passthru is disabled 1 = passthru is enabled, (no encode/decode) When passthru is enabled and duplex = 10, the chip outputs speech packets containing 8 Khz PCM data obtained from the selected INIF rather than outputting compressed channel data. This might be useful for test purposes. When passthru is enabled and duplex = 01, the chip expects to receive speech packets rather than channel packets. The speech data is sent directly to the selected OUTIF.

0 SKEW	select whether skew control is enabled 0 = skew control is disabled 1 = skew control is enabled When skew control is disabled, the encoder always inputs 160 samples per frame or the decoder always outputs 160 samples per frame. If the I2S interface is selected then the sample timing is derived from the 8 kHz I2S frame signal. If the ADC/DAC/DMIC interface is selected then the sample timing is derived from an internal 8 kHz clock which is derived from an internal 80 kHz clock which is derived from the AMBE-4020™ 4 MHz clock input. The AMBE-4020™ will produce a 50 Hz OFRAME signal that rises on the 0th sample and falls on the 80th sample. The encoder outputs packets after the falling edge of OFRAME. The decoder deadline for receiving a complete packet occurs at the falling edge of OFRAME. When skew control is enabled, the number of samples input by the encoder or output by the decoder is dependent upon the IFRAME signal. The IFRAME signal is assumed to be a 50 Hz (±2.5%) signal which may not be synchronous with either the AMBE-4020™ 4 MHz clock input or the 8 kHz I2S framing signals. The AMBE-4020™ sets frame boundaries at each rising edge of IFRAME. The number of speech samples per frame is therefore 160 ± 2.5% or 160 ± 4 samples. When skew control is enabled, the AMBE-4020™ still produces an OFRAME output signal, but the frequency will equal that of IFRAME. OFRAME rises on the 0th sample and falls on the 80th sample, where the 0th sample occurs at the rising edge of IFRAME. The rising edge of OFRAME occurs 0-125 us after the rising edge of IFRAME.

Table 54 PKT_STARTCODEC Fields

Upon receiving this field the mode is changed from packet mode to codec mode. If DUPLEX/INIF/OUTIF specify that the I^2S interface is utilized, then CODEC_RESETn is set high and then the codec configuration words that were set using the PKT_I2CDATA field, are sent via the I^2C pins.

After entering encoder codec mode (DUPLEX=DUPLEX_ENC), the AMBE-4020™ Vocoder Chip will output packets containing channel data every 20 ms. The channel data is obtained by encoding the speech samples received from the selected codec interface.

After entering decoder codec mode (DUPLEX=DUPLEX_DEC), the AMBE-4020™ Vocoder Chip expects to receive packets containing channel data every 20 ms.

After entering full duplex codec mode (DUPLEX=DUPLEX_FULL), the AMBE-4020™ Vocoder Chip, both the encoder and decoder are functional

After entering push-to-talk codec mode (DUPLEX=DUPLEX_PTT), the AMBE-4020™ Vocoder Chip monitors the state of the ENC/DEC pins to determine whether to encode or decode. When ENC is high then the encoder runs and channel packets are produced. When DEC is high but ENC is not, then the decoder runs and decodes received channel packets. When neither ENC nor DEC are high, then the AMBE-4020™ Vocoder Chip conserves power. The value of $(PMODE)^8$ specifies how power is conserved during PTT codec mode where $(PMODE)^8$ = 0x02 consumes the least amount of power.

After receiving this packet $(MODE)^8$ = 1, if the PASSTHRU bit is 0 or $(MODE)^8$ = 2, if the PASSTHRU bit is 1.

This field is only supported by control packets.

PKT_STOPCODEC field will switch the AMBE-4020™ Vocoder Chip from codec mode to packet mode and the codec reset signal is set low (in case I^2S codec interface was selected). After entering packet mode the AMBE-4020™ Vocoder Chip will stop outputting packets containing channel data every 20ms.

{ PKT_STOPCODEC = 0x2B } => { PKT_STOPCODEC, 0x00 }

This field is only supported by control packets.

PKT_STOPCODECF field will switch the AMBE-4020™ Vocoder Chip from codec mode to packet mode and the codec reset signal is set low (in case I^2S codec interface was selected). After entering packet mode the AMBE-4020™ Vocoder Chip will stop outputting packets containing channel data every 20ms.

{ PKT_STOPCODECF = 0x2C } => { PKT_STOPCODECF, 0x00 }

This packet field is supported by the AMBE-4020™ Full Duplex Vocoder Chip only and is slightly different than PKT_STOPCODEC. When using PKT_STOPCODECF, all channel packets are flushed out prior to sending the response packet to PKT_STOPCODECF. When using PKT_STOPCODEC some channel packets may be output after the response to PKT_STOPCODEC is received. Utilizing PKT_STOPCODECF rather than PKT_STOPCODEC can simplify exiting codec mode.

This field is only supported by control packets.

PKT_CHANFMT field will set the format of the channel packets output from the AMBE-4020™ Vocoder Chip.

{ PKT_CHANFMT = 0x15, $(CHANFMT)^{32}$ } => {PKT_CHANFMT, 0x00 }

This field is only supported by control packets.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved								ECHOFILT		PKT		LVL		ECHOSTAT	BER

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TONERCV		TONEDET		Reserved						SAMPLES		DSTATUS		ESTATUS	

Field	Description
31-24 Reserved	These values are reserved and should be set to 0
23-22 ECHOFILT	00 output channel packets never contain PKT_ECHOFILT field 01 output channel packets always contain PKT_ECHOFILT field with 2 coefficients per packet. 10 output channel packets always contain PKT_ECHOFILT field with 4 coefficients per packet. 11 output channel packets always contain PKT_ECHOFILT field with 8 coefficients per packet. ECHOFILT is only supported by AMBE-4020™ Full Duplex ECHOFILT is reserved (must set to 00) for AMBE-4020™ Half Duplex
21-20 PKT	00 output channel packets never contain PKT_PKT field 01 output channel packets always contain PKT_PKT field 10 every 10^{th} channel packet contains a PKT_PKT field 11 every 50^{th} channel packet contains a PKT_PKT field
19-18 LVL	00 output channel packets never contain PKT_LVL field 01 output channel packets always contain PKT_LVL field 10 every 10^{th} channel packet contains a PKT_LVL field 11 every 50^{th} channel packet contains a PKT_LVL field
17 ECHOSTAT	0 = output channel packets never contain PKT_ECHOSTAT field 1 = output channel packets always contain PKT_ECHOSTAT field

	ECHOSTAT is only supported by AMBE-4020™ Full Duplex ECHOSTAT is reserved (must set to 0) for AMBE-4020™ Half Duplex
16 BER	0 = output channel packets never contain PKT_BER field 1 = output channel packets always contain PKT_BER field
15-14 TONERCV	00 = output channel packets never contain PKT_TONERCV field 01 = output channel packets always contain PKT_TONERCV field 10 = output channel packets contain PKT_TONERCV field when received tone index changes 11 = output channel packets contain PKT_TONERCV field when a valid tone is received
13-12 TONEDET	00 = output channel packets never contain PKT_TONEDET field 01 = output channel packets always contain PKT_TONEDET field 10 = output channel packets contain PKT_TONEDET field when received tone index changes 11 = output channel packets contain PKT_TONEDET field when a valid tone is received
11-6 Reserved	These values are reserved and should be set to 0
5-4 SAMPLES	00 = output channel packets never contain PKT_SAMPLES field 01 = output channel packets always contain PKT_SAMPLES field 10 = output channel packets contains PKT_SAMPLES field whenever the number of samples changes 11 = output channel packets contain PKT_SAMPLES field whenever the number of samples is not 160
3-2 DSTATUS	00 = output channel packets never contain PKT_DSTATUS field 01 = output channel packets always contain PKT_DSTATUS field 10 = output channel packets contain PKT_DSTATUS field only when DSTATUS changes 11 = reserved
1-0 ESTATUS	00 = output channel packets never contain PKT_ESTATUS field 01 = output channel packets always contain PKT_ESTATUS field 10 = output channel packets contain PKT_ESTATUS field only when ESTATUS changes 11 = reserved

Table 55 PKT_CHANFMT Fields

PKT_SPCHFMT field will set the format of the speech packets output from the AMBE-4020™ Vocoder Chip

{ PKT_SPCHFMT = 0x16, $(SPCHFMT)^{32}$ } => {PKT_SPCHFMT, 0x00 }

This field is only supported by control packets.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
reserved					BER	TONERCV		reserved						DSTATUS	

Field	Description
31-11 Reserved	These values are reserved and should be set to 0
10 BER	0 = output speech packets never contain PKT_BER field 1 = output speech packets always contain PKT_BER field
15-14 TONERCV	00 = output speech packets never contain PKT_TONERCV field 01= output speech packets always contain PKT_TONERCV field 10 = output speech packets contain PKT_TONERCV field when received tone index changes

	11 = output speech packets contain PKT_TONERCV field when a valid tone is received
7-2 Reserved	These values are reserved and should be set to 0
1-0 DSTATUS	00 = output speech packets never contain PKT_DSTATUS field 01 = output speech packets always contain PKT_DSTATUS field 10 = output speech packets contain PKT_DSTATUS field only when DSTATUS changes 11 = reserved

Table 56 PKT_SPCHFMT Fields

PKT_PRODID field will cause the AMBE-4020™ Vocoder Chip to respond with a null-terminated string that contains product identification information. The maximum length of the returned sting is 16 bytes including the null character.

{ PKT_PRODID = 0x30 } => { PKT_PRODID, (PRODID)^{NULLSTRING} }

Half-Duplex Vocoder Chip Example Response:
{0x30 } => { 0x30, 'A', 'M', 'B', 'E', '4','0', '2', '0', 0x00 } (the string "AMBE4020" is returned)
Full-Duplex Vocoder Chip Example Response:
{0x30 } => { 0x30, 'A', 'M', 'B', 'E', '4','0', '2', '0', 'F', 'D', 0x00 } (the string "AMBE4020Full-Duplex" is returned)

This field is only supported by control packets.

PKT_VERSTRING field will cause the AMBE-4020™ Vocoder Chip to respond with a string that contains the product version number and date. The maximum length of the returned string is 32 bytes including the null character.

{ PKT_VERSTRING = 0x31 } => { PKT_VERSTRING, (VER)^{NULLSTRING} }

Example:
{0x31} => {0x31, "Release R00101 08-14-14", 0x00 } (is returned)

This field is only supported by control packets.

PKT_READY field (1 byte total) a packet containing this field is output by the AMBE-4020™ Vocoder Chip after a hard reset or soft reset (using a PKT_RESET field). After reset, the AMBE-4020™ outputs this packet when it is ready to receive packets. When the AMBE-4020™ is in sleep mode, it will send a packet containing the PKT_READY field upon receiving either a wake byte via UART_RX or upon any transition of UART_CTS. Note that a wake byte is discarded by the AMBE-4020™.

This field only occurs in the control response packet that is output after reset.

(hardware reset) => {PKT_READY = 0x39 }

(wake from sleep mode) => {PKT_READY = 0x39 }

PKT_ERROR field (2 bytes total) a packet containing this field is output by the AMBE-4020™ Vocoder Chip after an error condition occurs. The field only occurs in control response packets after an error condition is detected.

(error condition) => {PKT_ERROR = 0xFF, (ERRCODE)^{8} }

(ERRCODE)^{8}	Error Name	Description

1	INVALID_HEADER	Sent when it was expected to receive START_BYTE = 0x61, but something else was received.
2	RX_OVERFLOW	Sent when the packet receive queue overflowed, because the user ignored flow control status.
3	INVALID_PARITY	Sent in response to a packet with an invalid parity byte if parity fields are enabled.
4	MISSING_PARITY	Sent in response to a packet with a missing parity field if parity fields are enabled.

Table 57 PKT_ERROR Names

PKT_RESET field will cause the AMBE-4020™ Vocoder Chip to be reset. As a result, the AMBE-4020™ Vocoder Chip will lose all prior configuration settings and reset itself to the default power up state. Note that the AMBE-4020™ Vocoder Chip will re-read the boot pins.

{ PKT_RESET = 0x33 } => (reset) => {PKT_READY = 0x39}

The PKT_RESET field does not return a Response field; however, the AMBE-4020™ Vocoder Chip does output a PKT_READY packet after every reset (including both hard resets and packet resets). The PKT_READY packet can therefore be viewed as a response packet to the packet containing a PKT_RESET field.

This field is only supported by control packets.

PKT_PARITYMODE field can be used to enable or disable parity fields at the end of every packet.

{ PKT_PARITYMODE = 0x3F,(PARITYMODE)^8 } => { PKT_PARITYMODE, 0x00 }

If (PARITYMODE)^8 is 0 then parity fields will be disabled for all output packets beginning with the response to this packet. The AMBE-4020™ Vocoder Chip will not require a valid parity byte for future received packets.

If (PARITYMODE)^{8}is 1 then parity fields will be enabled for all output packets beginning with the response to this packet. The AMBE-4020™ Vocoder Chip will reject all future received packets that do not have a valid parity field.

All other values for (PARITYMODE)^8 are reserved and not to be used.

This field is only supported by control packets.

PKT_WRITEI2C field writes to an I^2C device such as a codec.

{ PKT_WRITEI2C = 0x44, $(NW)^8$, $(ADDR)^8$, $(REG)^8$, $(D_0)^8$, ... $(D_{NW-1})^8$ } => { PKT_WRITEI2C, 0x00 }

$(NW)^8$ specifies the number of registers that will be written to on the I2C slave device. $(ADDR)^8$ specifies the slave address of the I2C device, where the address is contained in the 7 MSBs and the LSB must always be 0. $(REG)^8$ specifies the number of the slave device register which will be written first. For slave devices that support it, additional writes ($(NW)^8 > 1$) will write to following registers on the slave device. $(D_0)^8$, ... $(D_{NW-1})^8$ specify the data to be written into the slave device register(s).

When the AMBE-4020™ receives this packet field it will transact the following sequence on the I2C bus (I2C_SCL/I2C_SCL):

1. Start Bit
2. byte containing slave address (7 MSBs) with R/W=0 (LSB)
3. byte containing register number
4. $(NW)^8$ bytes containing the data to be written into the slave device's register(s).
5. Stop Bit.

Note that the parameter $(I2CDIV)^8$ can be configured via the PKT_CONFIG_I2C field to select the I2C_SCL clock rate used during the above transfer. After the I2C transaction is complete, a response packet (UART) is sent.

This field is only supported by control packets.

PKT_READI2C field reads from an I^2C device such as a codec.

{ PKT_READI2C = 0x45, $(NR)^8$, $(ADDR)^8$, $(REG)^8$ } => { PKT_READI2C, $(NR)^8$, $(D_0)^8$, ... $(D_{NW-1})^8$ }

$(NR)^8$ specifies the number of registers that will be read from on the I2C slave device. Not all devices support $(NR)^8 > 1$. $(ADDR)^8$ specifies the slave address of the I2C device, where the address is contained in the 7 MSB's and the LSB must always be 0. $(REG)^8$ specifies the number of the slave device register which will be read from first. After reading the specified register(s), the data is sent back in a response packet. The response packet contains the number of bytes that were read followed by the data.

When the AMBE-4020™ receives this packet field, it will transact the following sequence on the I2C bus (I2C_SCL/I2C_SCL):

1. Start Bit
2. byte containing slave address (7 MSBs) with R/W=0 (LSB)
3. byte containing register number
4. repeated start
5. byte containing slave address (7 MSBs) with R/W=1 (LSB)
6. $(NR)^8$ bytes are read from the slave device.
7. Stop Bit.

Note that the parameter $(I2CDIV)^8$ can be configured via the PKT_CONFIG_I2C field to select the I2C_SCL clock rate used during the above transfer. After the I2C transaction is complete, a response packet (UART) is sent which contains the data read from the I2C slave device.

This field is only supported by control packets.

PKT_CONFIGI2C configures I^2C settings of the AMBE-4020™.

{ PKT_CONFIGI2C = 0x43, $(I2CADDR)^8$, $(I2CDELAY)^8$, $(I2CDIV)^8$ } => { PKT_CONFIGI2C, 0x00 }

There are two methods for configuring I2C devices.

The first method is selected by sending PKT_I2CDATA with $(NR)^8 > 0$ and then sending PKT_STARTCODEC where $(CODECCFG)^8$ is used to select the I2S codec interface. When these conditions are met, then a series of I2C write transactions are sent to the slave device, after receiving PKT_STARTCODEC. After receiving PKT_STARTCODEC but prior to I2C transactions, the CODEC_RESETn signal is set high (to take the I2C slave device out of reset). After CODEC_RESETn is set high, the AMBE-4020 waits for $10 \times (I2CDELAY)^8$ µs prior to beginning the I2C transaction. This allows time after reset in which the device is not yet ready to receive I2C data. $(I2CADDR)^8$ specifies the address of the slave device and $(I2CDIV)^8$ specifies the I2C clock rate where the default is 50 KHz. This is the default method for I2C configuration. It works well when 16 or less registers need to be configured and a single I2C slave device is addressed.

The second method uses PKT_WRITEI2C to provide more direct control of what is written to I2C device(s). When this method is used, then PKT_I2CDATA should be sent to configure $(NR)^8 = 0$. This will prevent the I2C transactions of the first method from taking place. If this method is used, then there is no effect on the CODEC_RESETn signal, however PKT_SET_CODECRESET/PKT_CLRCODECRESET can be used for that purpose if desired. $(I2CADDR)^8$ and $(I2CDELAY)^8$ have no effect on PKT_WRITEI2C/PKT_READI2C, however, $(I2CDIV)^8$ may still be used to specify the I2C clock rate. This method may be preferred to the first method if there are multiple slaves that need to be configured or if there is

a device that needs to have more than 16 registers configured. Another benefit of this method is that read back from the slave is permitted.

This field is only supported by control packets.

PKT_CLRCODECRESET field sets the codec reset signal to low.

{ PKT_CLRCODECRESET = 0x46 } => { PKT_CLRCODECRESET, 0x00 }

This field is only supported by control packets.

PKT_SETCODECRESET field sets the codec reset signal to high.

{ PKT_SETCODECRESET = 0x47 } => { PKT_SETCODECRESET, 0x00 }

This field is only supported by control packets.

PKT_DISCARDNCODEC field specifies the number of codec samples that are discarded when the codec interface is started. Default is 0.

{ PKT_DISCARNCODEC = 0x48, $(DISCARD)^{16}$ } => { PKT_DISCARDNCODEC, 0x00 }

This field is only supported by control packets.

PKT_DELAYNUS field specifies the amount of delay in microseconds prior to processing the next control field.

{ PKT_DELAYNUS = 0x49, $(DELAYUS)^{16}$ } => { PKT_DELAYNUS, 0x00 }

This field is only supported by control packets.

PKT_ERRTHRESH field allows the default error mitigation thresholds to be adjusted for vocoder rates that support FEC.

{ PKT_ERRTHRESH = 0x1A, $(SDDIST)^{16}$, $(HDERRS)^{16}$ } => { PKT_ERRTHRESH, 0x00 }

This field is only supported by control packets.

PKT_BOOTCFG field where $(WBOOT)^8$ = 1, 2, or 3 sets the EEPROM boot configuration 1, 2, or 3 data.

{ PKT_BOOTCFG = 0x0C, $(WBOOT)^8$, $(W_0)^8$ ... $(W_{127})^8$ } => { PKT_BOOTCFG, 0x00 }

The bytes of each boot configuration specify the initial value after reset for various different parameters as shown in Table 11: Parameters Specified for each Boot Configuration . For factory default values see Table 12: Factory Settings for each Boot Configuration. This packet field allows the user to program custom boot configurations into persistent memory. With a custom boot configuration, it is possible for users to boot the chip up with settings that are sufficient to their application. This reduces the time required to boot up and configure the AMBE-4020™ after reset. Boot configurations 1, 2 and 3 may be customized whereas boot configuration 0 is fixed.

This field is only supported by control packets.

Bytes		Parameter Specified	Description
W_0		Reserved	This byte must be 0
W_1,W_2, W_3		$(BAUD)^{24}$	Specify UART baud rate
W_4,W_5, W_6, W_7		$(SPCHFMT)^{32}$	Specifies format for outgoing speech packets
W_8,W_9, W_{10}, W_{11}		$(CHANFMT)^{32}$	Specifies format for outgoing channel packets
W_{12},W_{13}		$(ECONTROL)^{16}$	Specifies encoder control flags
W_{14},W_{15}		$(DCONTROL)^{16}$	Specifies decoder control flags
W_{16},W_{17}		$(DISCARD)^{16}$	Specifies the number of initial samples to discard
W_{18},W_{19}		$(GDIV)^{16}$	Specifies divider used to generate GFRAME
W_{20}		$(COMPAND)^8$	Specifies whether companding is used and choose A-law or μ-law
W_{21}		$(RATET)^8$	Specifies vocoder rate
W_{22}		$(CODECCFG)^8$	Selects codec mode duplex, interfaces, passthru, skew
W_{23}		$(FLOWPKT)^8$	Specifies the max number of packets before UART_RTS is set.
W_{24}		$(NCHANPKT)^8$	Specifies the number of channel packets to buffer in codec mode.
W_{25}		$(IGAIN)^8$	Specifies gain applied during encoder
W_{26}		$(OGAIN)^8$	Specifies gain applied during decoder
W_{27}		$(UFRAME\_HI)^8$	Specifies when UFRAME rises relative to rising edge of OFRAME.
W_{28}		$(UFRAME\_LO)^8$	Specifies when UFRAME falls relative to rising edge of OFRAME.
W_{29}		$(BREAKF)^8$	Specifies what happens when a UART break signal is received
W_{30}		$(MODE)^8$	Specifies Packet Mode or Codec Mode
W_{31}		$(PARITYMODE)^8$	Specifies whether the chip outputs parity fields in outgoing packets and checks for parity fields in incoming packets.
W_{32}		$(PMODE)^8$	Specifies power mode
W_{33}		reserved	Reserved bytes must be 0
W_{34} - W_{45}		$(RCW_0)^{16}$ - $(RCW_5)^{16}$	Specifies custom rate words used when $(RATET)^8 = 64$
W_{46}		$(I2CADDR)^8$	Specifies I2C slave address
W_{47}		$(I2CDELAY)^8$	Specifies delay after codec is reset, before I2C registers are written
W_{48}		$(I2CDIV)^8$	Dividers control for configuration of I2C clock rate
W_{49}		$(I2CNREG)^8$	Specifies the number of registers to be written to via I2C
W_{50} - W_{81}		$(I2CDATA_0)^8$ - $(I2CDATA_{31})^8$	Specifies register number/value pairs for each register to be written.
AMBE-4020™ Half-Duplex	W_{82} - W_{127}	Reserved	Reserved bytes must be 0
AMBE-4020™ Full-Duplex	W_{82}	Reserved	Reserved byte must be 1
	W_{83}	$(ECHOSUPLIM)^8$	Specifies the maximum attenuation for the echo suppressor
	W_{84},W_{85}	$(ECHOLEN)^{16}$	Specifies the length of the echo canceller filter in samples
	W_{86} – W_{127}	Reserved	Reserved bytes must be 0

Table 58 PKT_BOOTCFG

PKT_BOOTCFG field where $(RBOOT)^8$ is 5, 6, or 7 reads EEPROM boot configuration 1, 2, or 3 and returns the data in a response packet.

{ PKT_BOOTCFG = 0x0C, $(RBOOT)^8$ } => { PKT_BOOTCFG, $(W_0)^8$ … $(W_{127})^8$ }

The response packet returns the 128-byte boot configuration data specified by $(RBOOT)^8$ - 4.

This field is only supported by control packets.

PKT_BAUD field is used to change the baud rate of the UART interface.

$28800 \leq (BAUD)^{24} \leq 750000$.

{ PKT_BAUD = 0x57, $(BAUD)^{24}$ } => { PKT_BAUD, 0x00 }

The response packet will be sent using the new baud rate. The device that transmits the packet containing this field must change its own baud rate after sending this packet but before receiving the response packet. The recommend way of accomplishing this is to use flow control. Prior to sending PKT_BAUD, the transmitting device should set the AMBE-4020™'s CTS signal, which will prevent the AMBE-4020™ from sending a response packet. Next, the packet containing PKT_BAUD should be sent to the AMBE-4020™. Next, the transmitting device should change its own baud rate to the new baud rate. Finally, the transmitting device should set the CTS signal low, which will allow the AMBE-4020™ to send the response packet (using the new baud rate).

Another approach that does not utilize hardware flow control is as follows. Send a packet containing a PKT_BAUD field immediately followed by a PKT_DELAYNUS field. Then the transmitting device must change its own baud rate and be ready to receive the response packet before the delay expires. Using hardware flow control is the preferred method.

This field is only supported by control packets.

PKT_GPIO is used for GPIO on pins GPIO0, GPIO1, GPIO2, or GPIO3.

{ PKT_GPIO = 0x5F, $(GPIOFUNC)^8$, $(GPIOMASK)^8$ } => { PKT_GPIO, $(GPIORET)^8$ }

$(GPIOFUNC)^8$ specifies one of 8 possible functions:

GPIO_ON (0x00): enable or disable GPIO on the specified pin(s).
GPIO_DIR (0x01): set the direction of pin(s) to the direction(s) specified.
GPIO_VAL (0x02): set the specified pin(s) to the value(s) specified.
GPIO_SET (0x03): set specified pin(s) high.
GPIO_CLR (0x04): set specified pin(s) low.
GPIO_TOG (0x05): toggle specified pin(s).
GPIO_GET (0x06) read the state of specified pin(s).
GPIO_WAIT (0x07) wait for the pin(s) to be in a specified state.
GPIO_WAITN (0x08) wait for the pin(s) to not be in the specified state.

$(GPIOMASK)^8$ is broken down into 8 single-bit fields.

7	6	5	4	3	2	1	0
SEL_{GPIO3}	SEL_{GPIO2}	SEL_{GPIO1}	SEL_{GPIO0}	VAL_{GPIO3}	VAL_{GPIO2}	VAL_{GPIO1}	VAL_{GPIO0}

The SEL_{GPION} bits select which GPIO pin(s) the specified function is to be applied to. When SEL_{GPION} = 1, the function is applied to pin GPION, otherwise the function is not applied to GPION. Any function can be applied to one or more pins simultaneously. For function GPIO_ON, VAL_{GPION} specifies whether GPION is enabled (1) or disabled (0). For function GPIO_DIR, VAL_{GPION} specifies whether GPION is an input (0) or an output (1). For function GPIO_VAL, VAL_{GPION} specifies whether GPION is set low (0) or high (1). For function GPIO_WAIT, VAL_{GPION} specifies that the wait ends when GPION is low (0) or high (1). For functions GPIO_SET, GPIO_CLR, GPIO_TOG, and GPIO_GET, VAL_{GPION} is ignored.

GPIO Function	Description
GPIO_ON (0x00)	If SEL_{GPIO3} = 1, then GPIO3 is enabled if VAL_{GPIO3} = 1 or disabled if VAL_{GPIO3} = 0. If SEL_{GPIO2} = 1, then GPIO2 is enabled if VAL_{GPIO2} = 1 or disabled if VAL_{GPIO2} = 0. If SEL_{GPIO1} = 1, then GPIO1 is enabled if VAL_{GPIO1} = 1 or disabled if VAL_{GPIO1} = 0. If SEL_{GPIO0} = 1, then GPIO0 is enabled if VAL_{GPIO0} = 1 or disabled if VAL_{GPIO0} = 0. GPIORET=0
GPIO_DIR (0x01)	If SEL_{GPIO3} = 1, then GPIO3 is an output if VAL_{GPIO3} = 1 or an input if VAL_{GPIO3} = 0.

	If SEL_{GPIO2} = 1, then GPIO2 is an output if VAL_{GPIO2} = 1 or an input if VAL_{GPIO2} = 0. If SEL_{GPIO1} = 1, then GPIO1 is an output if VAL_{GPIO1} = 1 or an input if VAL_{GPIO1} = 0. If SEL_{GPIO0} = 1, then GPIO0 is an output if VAL_{GPIO0} = 1 or an input if VAL_{GPIO0} = 0. GPIORET=0
GPIO_VAL (0x02)	If SEL_{GPIO3} = 1, then GPIO3 = VAL_{GPIO3}. If SEL_{GPIO2} = 1, then GPIO2 = VAL_{GPIO2}. If SEL_{GPIO1} = 1, then GPIO1 = VAL_{GPIO1}. If SEL_{GPIO0} = 1, then GPIO0 = VAL_{GPIO0}. GPIORET=0
GPIO_SET (0x03)	If SEL_{GPIO3} = 1, then GPIO3 = 1. If SEL_{GPIO2} = 1, then GPIO2 = 1. If SEL_{GPIO1} = 1, then GPIO1 = 1. If SEL_{GPIO0} = 1, then GPIO0 = 1. GPIORET=0
GPIO_CLR (0x04)	If SEL_{GPIO3} = 1, then GPIO3 = 0. If SEL_{GPIO2} = 1, then GPIO2 = 0. If SEL_{GPIO1} = 1, then GPIO1 = 0. If SEL_{GPIO0} = 1, then GPIO0 = 0. GPIORET=0
GPIO_GET (0x06)	If SEL_{GPIO3} = 1, then $GPIORET_3$ = GPIO3. If SEL_{GPIO2} = 1, then $GPIORET_2$ = GPIO2. If SEL_{GPIO1} = 1, then $GPIORET_1$ = GPIO1. If SEL_{GPIO0} = 1, then $GPIORET_0$ = GPIO0.
GPIO_WAIT (0x07)	waits until all selected pins are all in the specified state, then sends response. GPIORET=0
GPIO_WAITN (0x08)	waits until all selected pins are not in the specified state, then sends response. GPIORET=0

Table 59 PKT_GPIO Function Description

This field is only supported by control packets.

PKT_UFRAME allows the rising edge and falling edge of the UFRAME pin to be adjusted relative to the rising edge of OFRAME. The UFRAME signal is always synchronous with the OFRAME signal. This packet allows the rising edge and falling edge of UFRAME to be customized.

{ PKT_UFRAME = 0x5E, $(UFRAME\_HI)^8$, $(UFRAME\_LO)^8$ } => { PKT_UFRAME, 0x00 }

OFRAME is generated by a counter within the AMBE-4020™.

When skew control is disabled, the counter continuously counts from 0 to 159. The time between counts is 125 us. When the counter is 0 this marks the start of a frame and the OFRAME signal is set high. When the counter reaches 80, OFRAME is set low.

When skew control is enabled, the counter is reset to 0 upon the rising edge of the IFRAME signal, and the OFRAME signal is set high within 125 us. The counter then increments at the nominal rate of once per 125 us. When the counter reaches 80, the OFRAME signal is set low. The total number of counts between two successive rising edges of IFRAME must be between 156 and 164.

After reset $(UFRAME\_HI)^8$ = 40 and $(UFRAME\_LO)^8$ = 120. This means that the UFRAME signal will be set high when the counter reaches 40 and it will be set low when the counter reaches 120. Therefore, by default the UFRAME signal is equivalent to the OFRAME signal delayed by 5 ms (40*125 us).

The rising edge and falling edge of UFRAME can be altered using this field, subject to the following restrictions:

$0 \le (\text{UFRAME\_HI})^8 < 160$ and $0 \le (\text{UFRAME\_LO})^8 < 160$ and $(\text{UFRAME\_HI})^8 \ne (\text{UFRAME\_LO})^8$ (if skew control is disabled)

$0 \le (\text{UFRAME\_HI})^8 < 156$ and $0 \le (\text{UFRAME\_LO})^8 < 156$ and $(\text{UFRAME\_HI})^8 \ne (\text{UFRAME\_LO})^8$ (if skew control is enabled)

UFRAME can be used by a microcontroller or other hardware to control the timing of other operations synchronously with the AMBE-4020™ packet rate.

This field is only supported by control packets.

PKT_BREAKF allows the function of the UART break signal to be specified.

{ PKT_BREAKF = 0x60, $(\text{BREAKF})^8$ } => { PKT_BREAKF, 0x00 }

$(\text{BREAKF})^8 = 0$, specifies that the break signal has no function and is ignored.

$(\text{BREAKF})^8 = 1$, specifies that the break signal causes reset. A packet containing a PKT_READY field will result. This is the default for boot configuration 0.

$(\text{BREAKF})^8 = 2$, specifies that the UART receiver is reset. This may be used to recover after sending invalid packets to the AMBE-4020™.

This field is only supported by control packets.

PKT_NCHANPKT is used to configure the number of received channel packets that are buffered during codec mode. It has no effect when processing speech using packet mode.

{ PKT_NCHANPKT = 0x5B, $(\text{NCHANPKT})^8$ } => { PKT_NCHANPKT, 0x00 }

$1 \le (\text{NCHANPKT})^8 \le 10$ specifies how many channel packets are buffered. By default after reset, $(\text{NCHANPKT})^8 = 1$ which is sufficient when the incoming channel packets are received at a constant 20 ms rate with very little jitter in the packet timing.

For situations where channel packets are received at irregular intervals, it may be necessary to increase the amount of buffering. Buffering can also be used to allow short bursts of a few channel packets. Note that increasing PKT_NCHANPKT implies increased worst case buffering delay.

In codec-mode, if the AMBE-4020™ maintains a buffer for received channel data. A frame is consumed by the decoder once every 20 ms. The time at which the decoder consumes a frame of channel data is referred to as the decoder "deadline" and this occurs at the falling edge of the OFRAME signal.

At the decoder deadline which occurs once every 20ms, the decoder does the following:
- If the decoder has more than $(\text{NCHANPKT})^8$ frames in the buffer it discards the oldest frame(s) until it has only $(\text{NCHANPKT})^8$ frames buffered.
- If the decoder has no frames in the buffer then it fills the gap using a frame" repeat", which is usually not noticeable.
- If the decoder has between 1 and $(\text{NCHANPKT})^8$ frames in its buffer then it decodes the oldest frame and removes it from the buffer.

Note that $(\text{FLOWPKT})^8$ as set by PKT_FLOWPKT must be greater than or equal to $(\text{NCHANPKT})^8$.

This field is only supported by control packets.

PKT_GDIV enables generation of the GFRAME signal and specifies the frequency of the signal.

{ PKT_GDIV = 0x5C, $(GDIV)^{16}$ } => { PKT_GDIV, 0x00 }

The AMBE-4020™ can generate a frame signal on the GFRAME pin. When $(GDIV)^{16} = 0$, no clock is generated on the GFRAME pin. When $(GDIV)^{16} = 625$, a 50 Hz frame signal is generated on the GFRAME pin.

The frequency of GFRAME is 31250 / $(GDIV)^{16}$ Hz.

GFRAME may be connected to IFRAME when codec mode is used with skew control enabled.

GFRAME may be useful during packet mode for generating a 50 Hz clock. This clock can be used by a connected micro-controller to establish the proper frame rate.

This field is only supported by control packets.

PKT_FLOWPKT specifies the maximum number of packets that can be received before UART_RTS is set high to stop receipt of additional packets.

{ PKT_FLOWPKT = 0x4E, $(FLOWPKT)^{8}$ } => { PKT_FLOWPKT, 0x00 }

The default after reset is $(FLOWPKT)^{8} = 3$, which means that if the AMBE-4020™ has received 3 or more unconsumed packets in its buffer, it will set RTS high. Each time the AMBE-4020™ consumes a packet, it checks to see if the number of packets has been reduced below the threshold, and if it has then RTS is set low again.

This field is only supported by control packets.

PKT_ECHOLEN specifies the length of the echo canceller filter in samples.

{ PKT_ECHOLEN = 0x64, $(ECHOLEN)^{16}$ } => { PKT_ECHOLEN, 0x00 }

The default after reset is $(ECHOLEN)^{8}$ =128 for all boot configurations, however boot configurations 1-3 are user programmable. $8 \leq (ECHOLEN)^{16} \leq 128$ and $(ECHOLEN)^{16}$ must be evenly divisible by 8. $(ECHOLEN)^{16}$ / 8 is the length of the echo canceller filter in milliseconds. Using the lowest filter length possible will minimize power consumption.

The setting for $(ECHOLEN)^{16}$ only affects full duplex codec mode when the echo canceller is enabled.

When full duplex codec mode is entered and the ADC/DAC interface is chosen, the filter length is reduced to 120 samples (15 milliseconds) if $(ECHOLEN)^{16} > 120$.

When full duplex codec mode is entered and the DMIC/DAC interface is chosen, the filter length is reduced to 96 samples (12 milliseconds) if $(ECHOLEN)^{16} > 96$.

This field is only supported by control packets.

PKT_PUTECHOFILT Sets the length of the echo canceller filter and specifies filter coefficients.

{ PKT_PUTECHOFILT = 0x65, $(ECHOLEN)^{16}$,$(EF_0)^{16}$,$(EF_1)^{16}$,.., $(EF_{(ECHOLEN)^{16}-1})^{16}$ } => { PKT_PUTECHOFILT, 0x00 }

The default after reset is $(EF_n)^{16} = 0$ for all n. Normally there is no need to use this field, however if it is expected that the echo canceller filter will be the same as a saved set of coefficients, then it is possible to start the echo canceller with a known set of coefficients which would reduce or eliminate the adaption time.

This field is only supported by control packets. This field is only supported by AMBE-4020™ Full Duplex.

PKT_GETECHOFILT Gets the echo canceller filter coefficients that were created by the echo canceller.

{ PKT_GETECHOFILT = 0x66 } => { PKT_GETECHOFILT, $(ECHOLEN)^{16}$, $(EF_0)^{16}$, $(EF_1)^{16}$, ..., $(EF_{(ECHOLEN)}{}^{16}-1)^{16}$ }

The default after reset is $(EF_n)^{16} = 0$ for all n. After exiting full duplex codec mode, this packet field may be sent to retrieve the adapted echo canceller filter coefficients.

This field is only supported by control packets. This field is only supported by AMBE-4020™ Full Duplex.

PKT_ECHOSUPLIM Specifies the maximum amount of suppression (in dB) applied by the echo suppressor.

{ PKT_ECHOSUPLIM = 0x69, $(ECHOSUPLIM)^8$ } => { PKT_ECHOSUPLIM, 0x00 }

The maximum amount of suppression is specified by variable $(ECHOSUPLIM)^8$. After reset, boot configuration 0-3 specify $(ECHOSUPLIM)^8 = 30$ by default. However boot configurations 1-3 may be reprogrammed by the user to select different reset behavior.

This field is only supported by control packets. This field is only supported by AMBE-4020™ Full Duplex.

PKT_PARITYBYTE field in an input packet is used to verify the validity of the entire packet. The $(PARITY)^8$ byte should be the exclusive-or of the entire packet excluding the START_BYTE. If this is not true, the chip will output a control packet containing PKT_ERROR. When used, the PKT_PARITYBYTE field must always be the last field in the packet.

When PKT_PARITYMODE is used to enable parity bytes, the final field in every output packet is the PKT_PARITYBYTE field.

{ PKT_PARITYBYTE = 0x2F, $(PARITY)^8$ } => { no response }

This field is supported in control packets, channel packets, and speech packets.

PKT_ECONTROL field specifies control flags used by the encoder. PKT_ECONTROL may be used to change the mode of the encoder prior to entering codec mode, or on a frame-by-frame basis while in codec mode. When received, the PKT_ECONTROL field will enable/disable various features of the encoder.

When input via a control packet:
{ PKT_ECONTROL = 0x05, $(ECONTROL)^{16}$ } => { PKT_ECONTROL, 0x00 }

When input via a channel packet or a speech packet:
{ PKT_ECONTROL=0x05, $(ECONTROL)^{16}$ } => { no response }

For $(ECONTROL)^{16}$, See Table 40 ECONTROL Flags. For example, to enable tone detection, DTX and noise suppression, PKT_ECONTROL data value would be 0x1840.

NOTE: PKT_ECONTROL and PKT_ESTATUS share the same Field ID Code (0x05). PKT_ECONTROL is used in input packets only, and PKT_ESTATUS is used only in output packets only.

This field is supported in control packets, channel packets, and speech packets.

PKT_DCONTROL field specifies control flags used by the decoder. PKT_DCONTROL may be used to change the mode of the decoder prior to entering codec mode, or on a frame-by-frame basis while in codec mode. When received, the PKT_DCONTROL field will enable/disable various features of the decoder.

When input via a control packet:
{ PKT_DCONTROL = 0x06, $(DCONTROL)^{16}$ } => { PKT_DCONTROL, 0x00 }

When input via a channel packet or a speech packet:
{ PKT_DCONTROL=0x06, $(DCONTROL)^{16}$ } => { no response }

For $(DCONTROL)^{16}$, See Table 44 DCONTROL Flags.

NOTE: PKT_DCONTROL and PKT_DSTATUS share the same Field ID Code (0x06). PKT_DCONTROL is used in input packets only, and PKT_DSTATUS is used only in output packets only.

This field is supported in control packets, channel packets, and speech packets.

PKT_CONTROL field provides an additional way to alter either $(ECONTROL)^{16}$ and $(DCONTROL)^{16}$.

{ PKT_CONTROL = 0x5D, $(FUNC)^{8}$, $(CONTROL)^{16}$ } => { PKT_CONTROL, 0x00 }

The LSB of $(FUNC)^{8}$ chooses whether $(ECONTROL)^{16}$ or $(DCONTROL)^{16}$ is modified
The remaining bits choose a function to be performed between the operand chosen by the LSB and $(CONTROL)^{16}$.

When $(FUNC)^{8} = 0$, $(ECONTROL)^{16} = (CONTROL)^{16}$
When $(FUNC)^{8} = 1$, $(DCONTROL)^{16} = (CONTROL)^{16}$
When $(FUNC)^{8} = 2$, $(ECONTROL)^{16} \mid= (CONTROL)^{16}$
When $(FUNC)^{8} = 3$, $(DCONTROL)^{16} \mid= (CONTROL)^{16}$
When $(FUNC)^{8} = 4$, $(ECONTROL)^{16}$ &= ~$(CONTROL)^{16}$
When $(FUNC)^{8} = 5$, $(DCONTROL)^{16}$ &= ~$(CONTROL)^{16}$

This field is supported by control packets, channel packets, and speech packets.

PKT_GAIN field can be used to set the input gain and output gain to anywhere between +20 and -20 dB. The default input gain and output gain are each 0 dB. The specified gain is applied in either packet mode or codec mode.

When input via a control packet:
{ PKT_GAIN = 0x4B, $(IGAIN)^{8}$, $(OGAIN)^{8}$ } => { PKT_GAIN, 0x00 }

When input via a channel packet:
{ PKT_GAIN = 0x4B, $(IGAIN)^{8}$, $(OGAIN)^{8}$ } => { no response }

$(IGAIN)^{8}$ is applied by the encoder. When $(IGAIN)^{8}$ is less-than 0 dB the encoder attenuates the signal. When $(IGAIN)^{8}$ is greater-than 0 dB the encoder amplifies the signal.

$(OGAIN)^{8}$ is applied by the decoder. When $(OGAIN)^{8}$ is less-than 0 dB the decoder attenuates the signal. When $(OGAIN)^{8}$ is greater-than 0 dB the decoder amplifies the signal.

It is recommended that the input and output gain are both 0 dB. Different values can be used for testing purposes.

This field is supported by control packets and channel packets.

PKT_ESTATUS fields are inserted within the outgoing channel packets when specified via PKT_CHANFMT.
PKT_ESTATUS fields are inserted within the outgoing speech packets when specified via PKT_SPCHFMT.

=> {PKT_ESTATUS = 0x05, $(ESTATUS)^{16}$ }

$(ESTATUS)^{16}$ contains status flags reported by the encoder. See Table 43 ESTATUS Flags.

NOTE: PKT_ECONTROL and PKT_ESTATUS share the same Field ID Code (0x05). PKT_ECONTROL is used in input packets only, and PKT_ESTATUS is used only in output packets only.
The field can be output in channel packets or speech packets.

PKT_DSTATUS fields are inserted within the outgoing channel packets when specified via PKT_CHANFMT.
PKT_DSTATUS fields are inserted within the outgoing speech packets when specified via PKT_SPCHFMT.

=> {PKT_DSTATUS = 0x06, $(DSTATUS)^{16}$ }

$(DSTATUS)^{16}$ contains status flags reported by the decoder. See Table 46 DSTATUS Flags.

NOTE: PKT_DCONTROL and PKT_DSTATUS share the same Field ID Code (0x06). PKT_DCONTROL is used in input packets only, and PKT_DSTATUS is used only in output packets only.

The field can be output in channel packets or speech packets.

PKT_TONERCV fields are inserted within the outgoing channel packets as specified by PKT_CHANFMT.
PKT_TONERCV fields are inserted within the outgoing speech packets as specified by PKT_SPCHFMT.

=> {PKT_TONERCV = 0x53, $(RCVIDX)^{8}$, $(RCVLVL)^{8}$ }

$(RCVIDX)^{8}$ reports the identity of a tone received by the decoder. See Table 60 TONE Index Values. $(RCVIDX)^{8}$ = 0xFF indicates that no tone was received.

$(RCVLVL)^{8}$ reports the level in dBm0 of a tone received by the decoder.

The field can be output in channel packets or speech packets.

PKT_BER fields are inserted within the outgoing channel packets as specified by PKT_CHANFMT.
PKT_BER fields are inserted within the outgoing speech packets as specified by PKT_SPCHFMT.

=> { PKT_BER = 0x56, $(BERLVL)^{16}$, $(DISTANCE)^{16}$,$(ERRORS)^{16}$ }

$(BERLVL)^{16}$ reports the bit error rate estimated by the FEC decoder.

$(DISTANCE)^{16}$ reports a soft-decision error metric for the current frame.

$(ERRORS)^{16}$ reports the number of bit errors in the current frame.

The field can be output in channel packets or speech packets.

PKT_TONEDET fields are inserted within the outgoing channel packets as specified by PKT_CHANFMT.

=> {PKT_TONEDET = 0x52, $(DETIDX)^8$, $(DETLVL)^8$ }

$(DETIDX)^8$ reports the identity of a tone detected by the encoder. See Table 60 TONE Index Values. $(DETIDX)^8$ = 0xFF indicates that no tone was detected.

$(DETLVL)^8$ reports the level in dBm0 of a tone detected by the encoder.

The field can be output in channel packets only.

PKT_ECHOFILT fields are inserted within the outgoing channel packets as specified by PKT_CHANFMT.

=> {PKT_ECHOFILT = 0x67, $(ECHOFILTIDX)^{16}$, $(EF_{IDX})^{16}$, ... $(EF_{IDX+N-1})$}

Where,
IDX = $(ECHOFILTIDX)^{16}$ & 0x3FFF
and
X = (($(ECHOFILTIDX)^{16}$ & 0xC000) >> 14;
and
$N = 2^X$

The field contains a partial output of the echo canceller filter coefficients. Either 2, 4, or 8 filter coefficients are reported in each PKT_ECHOFILT field depending upon what was specified using PKT_CHANFMT. The field may be useful to monitor the echo canceller. IDX is incremented and wrapped when it reaches $(ECHOLEN)^{16}$ such that the entire filter is scanned by successive PKT_ECHOFILT fields. PKT_CHANFMT specifies the filter scan rate (and therefore the size of this field) or specifies X=0 in which case this packet is disabled.

The field can be output during full duplex codec mode in channel packets only. This field is only supported by AMBE-4020™ Full Duplex.

PKT_ECHOSTAT fields are inserted within the outgoing channel packets as specified by PKT_CHANFMT.

=> {PKT_ECHOSTAT = 0x68, $(ATTENDB)^{16}$, $(ERLDB)^{16}$, $(LEVELDBM0)^{16}$ }

This field may be enabled in full duplex codec mode when the echo canceller is enabled to report echo canceller performance statistics.

$(ATTENDB)^{16}$ reports 100 times the echo canceller attenuation in dB. For example, if the echo canceller achieves 30.1 dB of attenuation then $(ATTENDB)^{16}$ = 3010. The attenuation reported is only accurate if the speech signal coming into the echo canceller contains a pure echo. The attenuation reported is just the ratio of the energy coming out of the echo canceller to the energy coming into the echo canceller.

$(ERLDB)^{16}$ reports 100 times the calculated echo return loss in dB. For example, if the ERL is 8.2 dB, then $(ERLDB)^{16}$ = 820. The ERL reported is only accurate if the speech signal coming into the echo canceller contains a pure echo. The ERL is just the ratio of the energy coming out of the decoder to the energy coming into the echo canceller. The echo canceller requires an ERL of more than 6 dB. Otherwise, the echo canceller filter will not adapt and echoes cannot be cancelled.

$(LEVELDBM0)^{16}$ reports the level of signal output by the echo canceller in dBm0 times 100. For example, if the signal level coming out of the echo canceller is -70.24 dBm0 then $(LEVELDBM0)^{16}$ = -7024.

The field can be output in channel packets only. This field is only supported by AMBE-4020™ Full Duplex.

PKT_LVL fields are inserted within the outgoing channel packets as specified by PKT_CHANFMT.

=> { PKT_LVL = 0x23, $(RXMAX)^{16}$, $(RXMIN)^{16}$, $(TXMAX)^{16}$, $(TXMIN)^{16}$ }

$(RXMAX)^{16}$ reports the maximum 16-bit linear PCM sample received by the encoder since the last time it was reported.

$(RXMIN)^{16}$ reports the minimum 16-bit linear PCM sample received by the encoder since the last time it was reported.

$(TXMAX)^{16}$ reports the maximum 16-bit linear PCM sample produced by the decoder since the last time it was reported.

$(TXMIN)^{16}$ reports the minimum 16-bit linear PCM sample produced by the decoder since the last time it was reported.

The field can be output in channel packets only.

PKT_PKT fields are inserted within the outgoing channel packets as specified by PKT_CHANFMT.

=> {PKT_PKT = 0x24, $(RXCNT)^{32}$, $(TXCNT)^{32}$, $(OVRFLW)^{32}$, $(SKIP)^{32}$, $(ERASE)^{32}$, $(RXMAXB)^{32}$, $(TXMAXB)^{32}$ }

$(RXCNT)^{32}$ reports the number of packets received since reset.

$(TXCNT)^{32}$ reports the number of packets transmitted since reset.

$(OVRFLW)^{32}$ reports the number of packet receive overflows since reset.

$(SKIP)^{32}$ reports the number of times that the decoder requested a frame repeat because it had 0 available channel packets in its buffer when the decoder deadline was encountered.

$(ERASE)^{32}$ reports the number of times that the decoder discarded a channel packet because it had more than TBD channel packets in its buffer when the decoder deadline was encountered.

$(RXMAXB)^{32}$ reports the maximum number of bytes in the receive buffer .

$(TXMAXB)^{32}$ reports the maximum number of bytes in the transmit buffer.

The field can be output in channel packets only.

PKT_SAMPLES fields are inserted within the outgoing channel packets as specified by PKT_CHANFMT.
PKT_SAMPLES fields may be inserted into channel packets input to the decoder to specify the number of samples synthesized by the decoder when operating in packet mode.

When output in channel packets:
=> {PKT_SAMPLES = 0x03, $(SAMPLES)^{8}$ }
$(SAMPLES)^{8}$ reports the number of samples (160 ± 4) in the last frame.
The packet is valid in either codec mode or packet mode.

When input in channel packets (packet mode only):
{PKT_SAMPLES = 0x03, $(SAMPLES)^{8}$ } => {no response}
$(SAMPLES)^{8}$ specifies that the output speech packet should contain $(SAMPLES)^{8}$ speech samples. The number of samples request must be between 156 and 164 samples. In packet mode, if an input channel packet does not contain a PKT_SAMPLES field then the number of samples produced by the decoder defaults to 160 for the current frame.

NOTE: if a channel packet containing this field is received while operating in codec mode, the field is ignored. In codec mode with skew control disabled, the number of samples produced by the decoder is always 160. In codec mode with skew control enabled, the number of samples produced by the decoder is dependent upon the IFRAME signal.

This field can be output in channel packets.
In packet mode, this field may be input in a channel packet.

PKT_TONEXMT fields can be used to force the encoder to transmit a tone frame. The frequency (or frequencies), amplitude, and duration of the tone are specified by this field. The specified tone will be transmitted regardless of what speech samples are input to the encoder.

{ PKT_TONEXMT=0x50, $(XMTIDX)^8$, $(XMTLVL)^8$, $(XMTDUR)^8$ }

$(XMTIDX)^8$ specifies the index of the tone to be transmitted. The index specifies the frequency or frequencies of the tone. See Table 60 TONE Index Values

$(XMTLVL)^8$ specifies the amplitude of the tone to be transmitted. The value is treated as an 8-bit signed value. The maximum tone amplitude is +3 dBm0 and the minimum tone amplitude is -90 dBm0. See Table 61 TONE Amplitude Values

$(XMTDUR)^8$ specifies the number of consecutive 20 ms frames that the tone will be transmitted. For example, $(XMTDUR)^8$ = 5, will specify that the tone duration is 100 ms. $(XMTDUR)^8$ = 255 specifies that the specified tone should be transmitted indefinitely until a new TONEXMT field is received with a duration that is not 255.

The field may be contained in either input channel packets or input speech packets.

Tone Index Values

Parameter Name	Description		TONE
	Frequency 1 (Hz)	Frequency 2 (Hz)	Index Value
Single Tones (The single tones span from 156.25 Hz to 3812.5 Hz in 31.25 Hz Increments)			
Single tone	156.25	N/A	0x05
	187.5	N/A	0x06
	218.75	N/A	0x07
	…	…	…
	…	…	…
	3812.5	N/A	0x7A
DTMF Tones			
1	1209	697	0x80
4	1209	770	0x81
7	1209	852	0x82
*	1209	941	0x83
2	1336	697	0x84
5	1336	770	0x85
8	1336	852	0x86
0	1336	941	0x87
3	1477	697	0x88
6	1477	770	0x89
9	1477	852	0x8A

#	1477	941	0x8B
A	1633	697	0x8C
B	1633	770	0x8D
C	1633	852	0x8E
D	1633	941	0x8F
KNOX Tones			
1	1052	606	0x90
4	1052	672	0x91
7	1052	743	0x92
*	1052	820	0x93
2	1162	606	0x94
5	1162	672	0x95
8	1162	743	0x96
0	1162	820	0x97
3	1297	606	0x98
6	1297	672	0x99
9	1297	743	0x9A
#	1297	820	0x9B
A	1430	606	0x9C
B	1430	672	0x9D
C	1430	743	0x9E
D	1430	820	0x9F
Call Progress			
Dial Tone	440	350	0xA0
Ring Tone	480	440	0xA1
Busy Tone	620	480	0xA2
Ring-UK	400	450	0xA3

Table 60 TONE Index Values

TONE Amplitude Values	
Description	TONE Amplitude Value
Max Amplitude Level = +3 dBm0	0x03
…	…
…	…
Min. Amplitude Level = -90 dBm0	0xA6

Table 61 TONE Amplitude Values

PKT_TONEGEN fields can be used to force the decoder to synthesize a tone frame. The frequency (or frequencies), amplitude, and duration of the tone are specified by this field. The specified tone will be synthesized by the decoder regardless of what channel data is input to the decoder.

{ PKT_TONEGEN = 0x51, $(GENIDX)^8$, $(GENLVL)^8$,$(GENDUR)^8$}

$(GENIDX)^8$ specifies the index of the tone to be generated. The index specifies the frequency or frequencies of the tone. See Table 60 TONE Index Values

$(GENLVL)^8$ specifies the amplitude of the tone to be generated. The value is treated as an 8-bit signed value. The maximum tone amplitude is +3 dBm0 and the minimum tone amplitude is -90 dBm0. See Table 61 TONE Amplitude Values

$(GENDUR)^8$ specifies the number of consecutive 20 ms frames that the tone will be generated. For example, $(GENDUR)^8 = 4$, will specify that the tone duration is 80 ms. $(GENDUR)^8 = 255$ specifies that the specified tone should be generated indefinitely until a new TONEGEN field is received with a duration that is not 255.

The field may only be contained in input channel packets.

PKT_CHAND fields within output channel packets contain compressed channel data bits from the encoder, packed 8 bits per byte. PKT_CHAND fields within input channel packets contain compressed channel data bits to be input to the decoder, packed 8 bits per byte:

For output channel packets in packet mode:
{speech packet input} => {encoder} => { PKT_CHAND = 0x01, $(NBITS)^8$, $(D_0)^8$ ... $(D_{NBYTES-1})^8$ }

For output channel packets in codec mode:
{speech samples in} => {encoder} => { PKT_CHAND = 0x01, $(NBITS)^8$, $(D_0)^8$ ... $(D_{NBYTES-1})^8$ }

For input channel packets in packet mode:
{ PKT_CHAND = 0x01, $(NBITS)^8$, $(D_0)^8$ ... $(D_{NBYTES-1})^8$ } => {decoder} => {speech packet output}

For input channel packets in codec mode:
{ PKT_CHAND = 0x01, $(NBITS)^8$, $(D_0)^8$ ... $(D_{NBYTES-1})^8$ } => {decoder} => {speech samples output}

For both input packets and output packets, $(NBITS)^8$ specifies the number of channel bits. NBYTES = (NBITS+7)/8 is the number of channel data bytes in the field.

Output channel packets always contain the PKT_CHAND field. For input channel packets, the field is optional.

This field is contained only in channel packets (either input or output).

PKT_CHAND4 fields within input channel packets contain compressed soft-decision channel data bits to be decoded with soft decision error correction enabled.

For input channel packets in packet mode:
{PKT_CHAND4 = 0x17, $(NBITS)^8$, $(D_0)^8$ ... $(D_{NBYTES-1})^8$ } => {decoder} => {speech packet output}

For input channel packets in codec mode:
{PKT_CHAND4 = 0x17, $(NBITS)^8$, $(D_0)^8$ ... $(D_{NBYTES-1})^8$ } => {decoder} => {speech samples output}

$(NBITS)^8$ specifies the number of channel bits. NBYTES = (NBITS+1)/2 is the number of channel data bytes in the field.

Each individual soft-decision bit is a 4-bit value where 0x0F is a most confident one and 0x00 is a most confident 0. The soft-decision bits are packed two bits per byte.

This field is contained only in input channel packets.

PKT_SPEECHD fields within input speech packets (packet mode) contain the speech data to be encoded. PKT_SPEECHD fields within output speech packets (packet mode) contain the decoded speech data. PKT_SPEECHD fields within input speech packets (decoder codec passthru mode) contain speech samples to be passed directly to the selected output codec interface. PKT_SPEECHD fields within output speech packets (encoder codec passthru mode) contain speech samples obtained directly from the selected input codec interface.

For input speech packets in packet mode:
{ PKT_SPEECHD=0x00, $(NSAMPLES)^8$, $(S_0,)^L$ ... $(S_{(NSAMPLES)-1})^L$ } => {encoder} => {channel packet output}

For output speech packets in packet mode:
{channel packet input} => {decoder} => { PKT_SPEECHD=0x00, $(NSAMPLES)^8$, $(S_0,)^L$ ... $(S_{(NSAMPLES)-1})^L$ }

For input speech packets in decoder codec passthru mode:
{ PKT_SPEECHD=0x00, $(NSAMPLES)^8$, $(S_0,)^L$ ... $(S_{(NSAMPLES)-1})^L$ } => {depacketize} => {speech sample output}

For output speech packets in encoder codec passthru mode:
{speech samples input} => {packetize} => { PKT_SPEECHD=0x00, $(NSAMPLES)^8$, $(S_0,)^L$ ... $(S_{(NSAMPLES)-1})^L$ }

In either input or output speech packets, $(NSAMPLES)^8$ specifies how many speech samples are contained in the packet.

When using 16 bit linear PCM Raw Speech data to be input to the encoder or output from the decoder there will be 16 bits per sample (L=16), this means at 160 samples there are 320 bytes of data. When using companded data (a-law or μ-law there are 8 bits of data per sample (L=8), this results in 160 bytes of data in 160 samples. The speech is denoted as S_0 thru $S_{NSAMPLES-1}$.

This field is contained only in speech packets (either input or output).

7.3.2 Control Packet Format

An input control packet (and the resulting response packet) uses the general packet format where the PACKET TYPE is equal to 0x00. Control packets can be used to configure the chip prior to operation and also to query for information from the chip. A control packet must contain one or more control fields. For each control packet received, the AMBE-4020™ Vocoder Chip sends back a response packet. The response packet for most fields just echoes back the control field identifier followed by a 0x00 byte to indicate that the control field was received successfully. For control fields that query for information, the response packet contains the requested information (1 or more bytes depending upon the control field identifier).

The control packet supports the following packet fields:

PKT_COMPAND	PKT_RESET	PKT_UFRAME
PKT_RATET	PKT_PARITYMODE	PKT_BREAKF
PKT_RATEP	PKT_WRITE I2C	PKT_NCHANPKT
PKT_INIT	PKT_READI2C	PKT_GDIV
PKT_PMODE	PKT_CONFIGI2C	PKT_FLOWPKT
PKT_I2CDATA	PKT_CLRCODECRESET	PKT_ECHOLEN
PKT_STARTCODEC	PKT_SETCODECRESET	PKT_PUTECHOFILT
PKT_STOPCODEC	PKT_DISCARDCODEC	PKT_GETECHOFILT
PKT_STOPCODECF	PKT_DELAYNUS	PKT_ECHOSUPLIM
PKT_CHANFMT	PKT_ERRTHRESH	PKT_PARITYBYTE
PKT_SPCHFMT	PKT_BOOTCFG	PKT_ECONTROL
PKT_PRODID	PKT_BOOTCFG	PKT_DCONTROL
PKT_VERSTRING	PKT_BAUD	PKT_CONTROL
PKT_READY	PKT_GPIO	PKT_GAIN

Table 62 Control Packet Fields

In general, one or more control packet fields may be grouped together in a single control packet. The response packet will then have a corresponding number of response fields. When parity bytes are enabled, the final field in the control packet must be a parity field and the final field in the response packet will also be a parity field. It does not make sense to group PKT_RESET with other fields because the resulting reset will invalidate the remaining fields. In addition, it is not recommended to combine PKT_BAUD with other fields other than PKT_PARITYBYTE.

When sending a control packet, it is recommended to wait for the response packet to be received (and verified) before sending the next control packet.

Symbolically, the control packet and its resulting response packet could be summarized as follows:

$[\{H_C\}\{F_0\}\{F_1\}\ldots\{F_{N-1}\}] \Rightarrow [\{H_R\}\{RF_0\}\{RF_1\}\ldots\{RF_{N-1}\}]$

Where,

1. $[\{H_C\}\{F_0\}\{F_1\}\ldots\{F_{N-1}\}]$ denotes the complete control packet sent to the AMBE-4020™.
2. $[\{H_R\}\{RF_0\}\{RF_1\}\ldots\{RF_{N-1}\}]$ denotes the complete response packet received in reply to the above control packet.
3. The total number of fields in both the control packet and the response packet is denoted N.
4. H_C and H_R are the four-byte headers (comprised of start byte, length byte, and type byte) of the control packet and its resulting response packet. Note that the type byte is 0x00 for the control packet and the response packet. The length bytes within H_C must be the total number of bytes contained in $F_0 \ldots F_{N-1}$. The length bytes within H_R must be the total number of bytes contained in $RF_0 \ldots RF_N$.
5. $F_0 \ldots F_{N-1}$ are one or more control packet fields, each one or more bytes in length depending upon the particular field identifier. The first byte in each field is the field identifier. Most fields require additional data following the field identifier.
6. $RF_0 \ldots RF_{N-1}$ are one or more response fields. The number of response packet fields matches the number of control packet fields and they occur in the same order. RF_n is the response field for F_n. The majority of control fields have a two-byte response that contains the field identifier followed by 0x00 (indicating "no error"). Some response fields such as PKT_PRODID and PKT_VERSTRING return data within the response field.

7.3.3 Input Speech Packet Format

An input speech packet uses the general packet format where the PACKET TYPE is equal to 0x02.

Input speech packets must not be used when the chip is operating in codec mode. In these case, input speech samples are acquired from the specified codec interface (either I2S, ADC, or DMIC).

In packet mode, for every speech packet input to the AMBE-4020™ Vocoder chip, the chip will output channel packet.

In decoder codec passthru mode, for every speech packet input to the AMBE-4020™ Vocoder chip, the samples are output directly on the selected codec interface (either I2S or DAC).

The following fields are supported within speech input packets:

PKT_PARITYBYTE	PKT_DCONTROL	PKT_TONEXMT
PKT_ECONTROL	PKT_CONTROL	PKT_SPEECHD

Table 63 Input Speech Packet Fields

However, in decoder codec passthru mode, only PKT_SPEECHD and PKT_PARITYBYTE fields are allowed.
All speech packets must contain exactly one PKT_SPEECHD field.

Symbolically, the input speech packet and the resulting channel packet can be summarized as follows:

$[\{H_S\}\{SF_0\}\{SF_1\}\ldots\{SF_{NS}\}] \Rightarrow [\{H_C\}\{CF_0\}\{CF_1\}\ldots\{CF_{NC}\}]$

Where,

1. $[\{H_S\}\{SF_0\}\{SF_1\}\ldots\{SF_{NS}\}]$ denotes the complete speech packet sent to the AMBE-4020™.
2. $[\{H_C\}\{CF_0\}\{CF_1\}\ldots\{CF_{NC}\}]$ denotes the complete channel packet received in reply to the above speech packet.
3. The total number of fields in the speech packet is denoted NS. The total number of fields in the channel packet is denoted NC. Note that this differs from control packets in that there is not a one-to-one relation between the fields in the speech packet to the fields in the resulting channel packet.

4. H_S and H_C are the four-byte headers (comprised of start byte, length byte, and type byte) of the input speech packet and its resulting channel packet. Note that the type byte is 0x02 for the speech packet and 0x01 for the channel packet. The length bytes within H_S must be the total number of bytes contained in SF_0 … $SF_{NS.}$ The length bytes within H_C must be the total number of bytes contained in CF_0 … $CF_{NC.}$
5. SF_0 … SF_{NS} are one or more speech packet fields. The first byte in each field is the field identifier. All speech packet fields require additional data following the field identifier. The first field, SF_0, must be a complete PKT_SPEECHD field. Additional fields are optional. If parity bytes are enabled, the final field, SF_{NS-1}, must be a valid PKT_PARITYBYTE field.
6. CF_0 … CF_{N-1} are one or more channel fields. Refer to Section Input Channel Packet Format for more information on the input channel packet format.

7.3.4 Output Speech Packet Format

When the AMBE-4020™ Vocoder chip is operating in packet mode, a speech packet (packet type 0x02) is output whenever the chip receives an input channel packet (packet type 0x01). The format of the output speech packet can be configured using the PKT_SPCHFMT control field.

In addition, when the AMBE-4020™ Vocoder chip is operating in encoder codec passthru mode, a speech packet (packet type 0x02) is output every 20 ms to transfer speech acquired directly from the specified input codec interface (either I2S, ADC, or DMIC). Output speech packets in encoder codec passthru mode contain a PKT_SPEECHD field and a PKT_PARITYBYTE field (if parity bytes are enabled). The PKT_SPCHFMT control field is irrelevant in this case.

Output speech packets may contain any of the following fields:

PKT_PARITYBYTE	PKT_TONERCV	PKT_SPEECHD
PKT_DSTATUS	PKT_BER	

Table 64 Output Speech Packet Fields

Symbolically, the input channel packet and the resulting speech packet can be summarized as follows:

$[\{H_C\}\{CF_0\}\{CF_1\}\ldots\{CF_{NC}\}] \Rightarrow [\{H_S\}\{SF_{SPEECHD}\}\{SF_{DSTATUS}\}\{SF_{TONERCV}\}\{SF_{BER}\}\{SF_{PARITY}\}]$

Where,

1. $[\{H_C\}\{CF_0\}\{CF_1\}\ldots\{CF_{NC}\}]$ denotes the complete channel packet sent to the AMBE-4020™. The total number of fields in the channel packet is denoted NC.
2. $[\{H_S\}\{SF_{SPEECHD}\}\{SF_{DSTATUS}\}\{SF_{TONERCV}\}\{SF_{BER}\}\{SF_{PARITY}\}]$ denotes the complete speech packet output in response to the incoming channel packet.
3. H_S and H_C are the four-byte headers (comprised of start byte, length byte, and type byte) of the input speech packet and its resulting channel packet. Note that the type byte is 0x02 for the speech packet and 0x01 for the channel packet. The length bytes within H_S must be the total number of bytes contained in SF_0 … $SF_{NS.}$ The length bytes within H_C must be the total number of bytes contained in CF_0 … $CF_{NC.}$
4. CF_0 … CF_{N-1} are one or more channel fields. Refer to Section Input Channel Packet Format for more information on the input channel packet format.
5. $\{SF_{SPEECHD}\}$ denotes a PKT_SPEECHD field in is always present in output speech packets.
6. $\{SF_{DSTATUS}\}$ denotes an optional PKT_DSTATUS field.
7. $\{SF_{TONERCV}\}$ denotes an optional PKT_TONERCV field.
8. $\{SF_{BER}\}$ denotes an optional PKT_BER field.
9. $\{SF_{PARITY}\}$ denotes an optional PKT_PARITYBYTE field.
10. Presence of $\{SF_{DSTATUS}\}$, $\{SF_{TONERCV}\}$ and $\{SF_{BER}\}$ fields is determined by the settings configured using the PKT_SPCHFMT control field. Depending upon the configuration, the fields may be present in some packets and absent in others. Presence of the $\{SF_{PARITY}\}$ field is dependent upon whether parity bytes are enabled or disabled.
11. The order of the output speech packet fields (when present) is always as shown above.

7.3.5 Input Channel Packet Format

An input channel packet uses the general packet format where the PACKET TYPE is equal to 0x01.

When operating in packet mode:
Every channel packet input (packet type 0x01) to the AMBE-4020™ Vocoder chip results in an output speech packet (packet type 0x02). The channel data is passed to the decoder for decoding and the resulting speech samples are output via the speech packet.

When operating in decoder codec mode, full-duplex codec mode, or push-to-talk codec mode with (ENC=0, DEC=1):
The channel data from the input channel packet is passed to the decoder and the resulting speech samples are sent to the selected codec interface (either I2S or DAC).

When operating in encoder codec mode:
Although no channel data is required, since the decoder is not running, input channel packets may be used to control the encoder using the following fields: PKT_ECONTROL, PKT_CONTROL, PKT_GAIN, PKT_TONEXMT.

The fields available for use in input channel packets are as follows:

PKT_PARITYBYTE	PKT_GAIN	PKT_CHAND
PKT_ECONTROL	PKT_SAMPLES	PKT_CHAND4
PKT_DCONTROL	PKT_TONEXMT	
PKT_CONTROL	PKT_TONEGEN	

Table 65 Input Channel Packet Fields

Symbolically, the input channel packet and the resulting speech packet can be summarized as follows:

$[\{H_C\}\{CF_0\}\{CF_1\}\ldots\{CF_{NC}\}] \Rightarrow [\{H_S\}\{SF_0\}\{SF_1\}\ldots\{SF_{NS}\}]$

Where,

1. $[\{H_C\}\{CF_0\}\{CF_1\}\ldots\{CF_{NC}\}]$ denotes the complete channel packet sent to the AMBE-4020™.
2. $[\{H_S\}\{SF_0\}\{SF_1\}\ldots\{SF_{NS}\}]$ denotes the complete speech packet received in reply to the above speech packet. The total number of fields in the speech packet is denoted NS. The total number of fields in the channel packet is denoted NC. Note that this differs from control packets in that there is not a one-to-one relation between the fields in the speech packet to the fields in the resulting channel packet.
3. H_C and H_S are the four-byte headers (comprised of start byte, length byte, and type byte) of the input channel packet and its resulting speech packet. Note that the type byte is 0x01 for the channel packet and 0x02 for the speech packet. The length bytes within H_S must be the total number of bytes contained in SF_0 … $SF_{NS.}$ The length bytes within H_C must be the total number of bytes contained in CF_0 … $CF_{NC.}$
4. SF_0 … SF_{NS} are one or more speech packet fields. Refer to Section Output Speech Packet Format for more information on the format of output speech packets.
5. CF_0 … CF_{N-1} are one or more channel packet fields chosen from Table 65 Input Channel Packet Fields. The order of the fields does not matter except that the PKT_PARITYBYTE field, if present, must be the last field.

When sending channel packets to the AMBE-4020™ in packet mode, it is not necessary to wait for the response packet to be received before sending the next packet, however it is necessary to utilize flow control to prevent overflowing the receive buffer. Queuing packets is often necessary in order to get adequate packet throughput.

When sending channel packets to the AMBE-4020™ in codec mode, several situations can be handled:

1. In the simplest case, the AMBE-4020™ receives one channel packet every 20 ms with very little or no jitter in the packet timing. In this case, no channel packet buffering is needed and $(NCHANPKT)^8 = 1$ is ideal.
2. If packets are received every 20 ms, but there is significant packet jitter or if packets are received in blocks then $(NCHANPKT)^8 > 1$ is warranted.

7.3.6 Output Channel Packet Format0

When the AMBE-4020™ Vocoder chip is operating in packet mode, a channel packet (packet type 0x01) is output whenever the chip receives an input speech packet (packet type 0x02). The format of the output channel packet can be configured using the PKT_CHANFMT control field.

In addition, when the AMBE-4020™ Vocoder chip is operating in either encoder codec mode, full-duplex codec mode, or push-to-talk codec mode with ENC=1, a periodic channel packet is output every 20 ms.

In addition, when the AMBE-4020™ Vocoder chip is operating in either decoder codec mode or push-to-talk codec mode with ENC=0 and DEC = 1, a periodic channel packet (with no PKT_CHAND field) may be produced if configured via a PKT_CHANFMT control field.

Channel packets output from the AMBE-4020™ support the following packet fields:

PKT_PARITYBYTE	PKT_BER	PKT_LVL
PKT_ESTATUS	PKT_TONEDET	PKT_PKT
PKT_DSTATUS	PKT_ECHOFILT	PKT_SAMPLES
PKT_TONERCV	PKT_ECHOSTAT	PKT_CHAND

Table 66 Output Channel Packet Fields

Symbolically, the input speech packet and the resulting channel packet can be summarized as follows:

$[\{H_S\}\{SF_0\}\{SF_1\}\ldots\{SF_{NS-1}\}] =>$
$[\{H_c\}\{CF_{CHAND}\}\{CF_{ESTATUS}\}\{CF_{DSTATUS}\}\{CF_{SAMPLES}\}\{CF_{TONEDET}\}\{CF_{TONERCV}\}\{CF_{BER}\}\{CF_{ECHOFILT}\}\{CF_{ECHOSTAT}\}\{CF_{LVL}\}\{CF_{PKT}\}\{CF_{PARITY}\}]$

Where,

1. $[\{H_S\}\{SF_0\}\{SF_1\}\ldots\{SF_{NS}\}]$ denotes the complete speech packet sent to the AMBE-4020™. The total number of fields in the speech packet is denoted NS.
2. $[\{H_c\}\{CF_{CHAND}\}\{CF_{ESTATUS}\}\{CF_{DSTATUS}\}\{CF_{SAMPLES}\}\{CF_{TONEDET}\}\{CF_{TONERCV}\}\{CF_{BER}\}\{CF_{ECHOFILT}\}\{CF_{ECHOSTAT}\}\{CF_{LVL}\}\{CF_{PKT}\}\{CF_{PARITY}\}]$ denotes the complete channel packet output in response to the incoming speech packet.
3. H_S and H_C are the four-byte headers (comprised of start byte, length byte, and type byte) of the input speech packet and its resulting channel packet. Note that the type byte is 0x02 for the speech packet and 0x01 for the channel packet. The length bytes within H_S must be the total number of bytes contained in SF_0 … $SF_{NS-1.}$ The length bytes within H_C must be the total number of bytes contained in CF_{CHAND} … $CF_{PARITY.}$
4. SF_0 … SF_{NS-1} are one or more input speech packet fields. Refer to Section Input Speech Packet Format for more information on the input speech packet format.
5. $\{CF_{CHAND}\}$ denotes a PKT_CHAND field in is always present in output speech packets (for packet mode).
6. $\{CF_{ESTATUS}\}$ denotes an optional PKT_ESTATUS field.
7. $\{CF_{DSTATUS}\}$ denotes an optional PKT_ESTATUS field.
8. $\{CF_{SAMPLES}\}$ denotes an optional PKT_SAMPLES field.
9. $\{CF_{TONEDET}\}$ denotes an optional PKT_TONEDET field
10. $\{CF_{TONERCV}\}$ denotes an optional PKT_TONERCV field.
11. $\{CF_{BER}\}$ denotes an optional PKT_BER field.
12. $\{CF_{ECHOFILT}\}$ denotes an optional PKT_ECHOFILT field.
13. $\{CF_{ECHOSTAT}\}$ denotes an optional PKT_ECHOSTAT field.
14. $\{CF_{LVL}\}$ denotes an optional PKT_LVL field.
15. $\{CF_{PKT}\}$ denotes an optional PKT_PKT field.
16. $\{CF_{PARITY}\}$ denotes an optional PKT_PARITYBYTE field.
17. Presence of $\{CF_{ESTATUS}\}$ through $\{CF_{PKT}\}$ fields is determined by the settings configured using the PKT_CHANFMT control field. Depending upon the configuration, the fields may be present in some packets and absent in others. Presence of the $\{CF_{PARITY}\}$ field is dependent upon whether parity bytes are enabled or disabled. In packet mode, the $\{CF_{CHAND}\}$ is always present. In codec mode, $\{CF_{CHAND}\}$ is present only if the encoder is running.
18. The order of the output channel packet fields (when present) is always as shown above.

7.4 Example Packets

7.4.1 Speech Packet Example 1

The simplest way to operate the AMBE-4020™ Vocoder Chip in packet mode is to send it a packet and then wait for a response packet. However, using this method, the vocoder is idle during the time when a packet is being received by the AMBE-4020™ Vocoder Chip and during the time in which the AMBE-4020™ Vocoder Chip is transmitting the response packet.

Following is an example speech packet (hexadecimal) for input to the AMBE-4020™ Vocoder Chip:

Speech Packet					
Header			SPEECHD Field		
StartByte	Length	Type	SPEECHD field identifier	SPEECHD No. of Samples	SPEECHD Data
61	0144	02	00	A0	0000000100020003000400050006000700080009000 A000B000C000D000E000F001000110012001300140 0 150001601700180019001A001B001C001D001E001F0 020002100220023002400250026002700280029002A 002B002C002D002E002F003000310032003300340033 500360037003800390 03A003B003C003D003E003F00 400041004200430044004500460047004800490 04A0 04B004C004D004E004F005000510052005300540055 005600570058005900 5A005B005C005D005E005F006 000610062006300640065006600670068006900 6A00 6B006C006D006E006F007000710072007300740075 0 076007700780079007A007B007C007D007E007F0080 008100820083008400850086008700880089008A008 B008C008D008E008F009000910092009300940095 00 960097009800990 09A009B009C009D009E009F

Table 67 Speech Packet Example 1

The first byte (0x61) is the packet header byte. The next two bytes (0x0144) specify the total length of the packet fields is 324 bytes. Note that the total packet length including the header, length, and type is 328 bytes. The next byte (0x02) specifies that the packet type is a speech packet. The following byte (0x00) specifies channel 0 for subsequent fields. The next byte (0x00) is a SPEECHD field identifier and the following byte (0xA0) tells the AMBE-4020™ Vocoder Chip that the SPEECHD Data field contains 160 speech samples, occupying 320 bytes. The final 320 bytes contain the speech samples. For this particular example, the speech samples increment from 0 to 159. Note that the MS byte of each sample is transmitted/received prior to the LS byte of each sample. This convention is used whenever a 16-bit number is contained in a packet.

7.4.2 Speech Packet Example 2

The following packet is another example of speech input

Speech Packet											
Header			SPEECHD Field			ECONTROL Field		TONEXMT Field			
StartByte	Length	Type	SPEECHD Field identifier	SPEECHD No. of Samples	SPEECHD Data	ECONTROL Field identifier	ECONTROL flags	TONEXMT Field identifier	TONEXMT Index Value	TONEXMT Amplifier Value	TONEXMT Duration
61	014A	02	00	A0	0000000100020003000400500060007000800900A000B000C000D000E000F0010001100120013001400150001600170018001900A001B001C001D001E001F0020002100220023002400250026002700280029002A002B002C002D002E002F0030003100320033003400350036003700380039003A003B003C003D003E003F0040004100420043004400450046004700480049004A004B004C004D004E004F0050005100520053005400550056005700580059005A005B005C005D005E005F0060006100620063006400650066006700680069006A006B006C006D006E006F0070007100720073007400750076007700780079007A007B007C007D007E007F0080008100820083008400850086008700880089008A008B008C008D008E008F009000910092009300940095009600970098009900A009B009C009D009E009F	05	0000	50	13	00	04

Table 68 Speech Packet Example 2

This is the similar to the prior example except that a PKT_ECONTROL field and a PKT_TONEXMT field were added to the end of the packet. The length field changed to 0x014a because the packet length increased by 6 bytes. For the new bytes at the end of the packet (0x05) is the PKT_ECONTROL field identifier. The following two bytes (0x0000) specifies that the encoder

control flags should be set to 0x0000. The next byte (0x50) is a TONEXMT field identifier. The next three bytes (0x03, 0x00, and 0x04) specify tone index of 19, a tone amplitude of 0 dBm0, and a tone duration of 80 ms.

7.4.3 Channel Packet Example 1

Following is an example channel packet (hexadecimal) for input to the AMBE-4020™ Vocoder Chip:

Channel Packet					
Header			CHAND Field		
StartByte	Length	Type	CHAND Field Identifier	CHAND No. of Bits	CHAND Data
61	000C	01	01	50	00112233445566778899

Table 69 Channel Packet Example 1

The first byte (0x61) is the packet header byte. The next two bytes (0x000C) specify that the length of the packet (excluding the header, length, and type bytes) is 12 bytes. The next byte (0x01) specifies that the packet type is a channel packet. The next byte (0x01) is the field identifier for a PKT_CHAND field. The next byte (0x50) specifies that 80 bits of channel data follow. The bits are packed 8 bits per byte such that the 80 bits are contained in the 10 bytes that follow. The final 10 bytes contain the channel data. The bits are output with the most significant (and most sensitive to bit-errors) bits in the first byte and the least significant (and least sensitive to bit-errors) bits in the last byte. For bit-rates that are not an even multiple of 400 bps, the MSBs of the last byte are used to hold the channel data, and the LSBs will be padded with zeros.

7.4.4 Channel Packet Example 2

Following is another example of a channel packet for input to the AMBE-4020™ Vocoder Chip:

Channel Packet									
Header			CHAND Field			SAMPLES Field		DCONTROL Field	
StartByte	Length	Type	PKT_CHAND Field Identifier	PKT_CHAND Number of Bits	CHAND Data	PKT_SAMPLES Field Identifier	PKT_SAMPLES Number of Samples	PKT_DCONTROL Field	PKT_DCONTROL Value
61	0010	01	01	38	00112233445566	03	A1	06	0000

Table 70 Channel Packet Example 2

The first byte (0x61) is the packet header byte. The next two bytes (0x0010), specify that the length of the packet (excluding the header, length, and type bytes) is 16 bytes. The next byte (0x01) specifies that the packet type is a channel packet. The next byte (0x01) is a PKT_CHAND specifier and the following byte (0x38) specifies that 56 bits (7 bytes) of channel data

follow. The next 7 bytes contain the channel data to be decoded by the decoder. The next byte (0x03), is a field identifier for a PKT_SAMPLES field. The next byte (0xA1), specifies that the decoder will output 161 samples rather than the normal 160 samples when it produces the resulting speech packet. The next byte (0x06), is the field identifier for a DCONTROL field. The final 2 bytes (0x0000), are used to control the decoder mode.

SECTION 8

8 Appendices

8.1 80-pin Low-Profile Quad Flat Pack (LQFP) Package Details

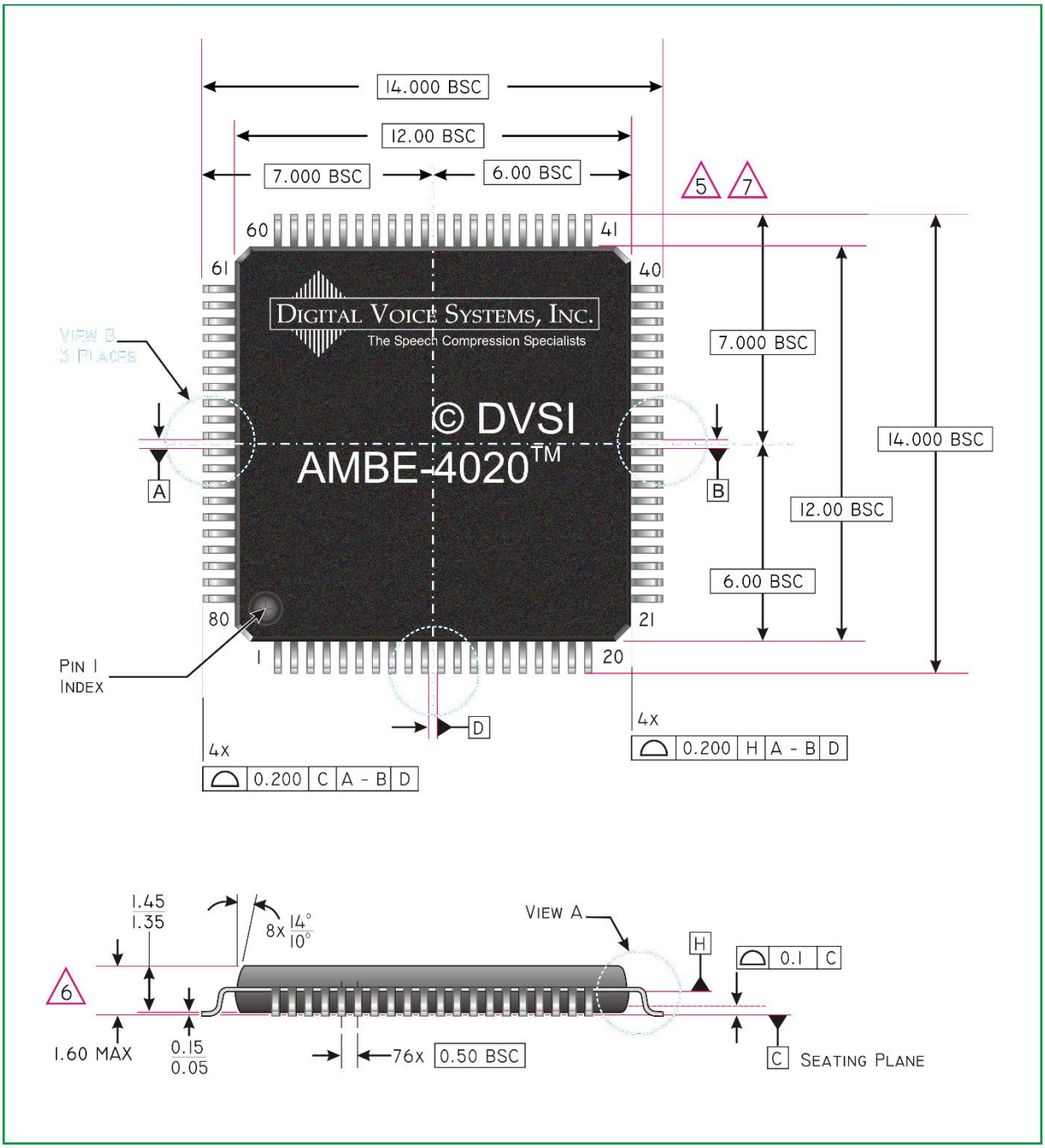


Figure 42 LQFP Mechanical Details

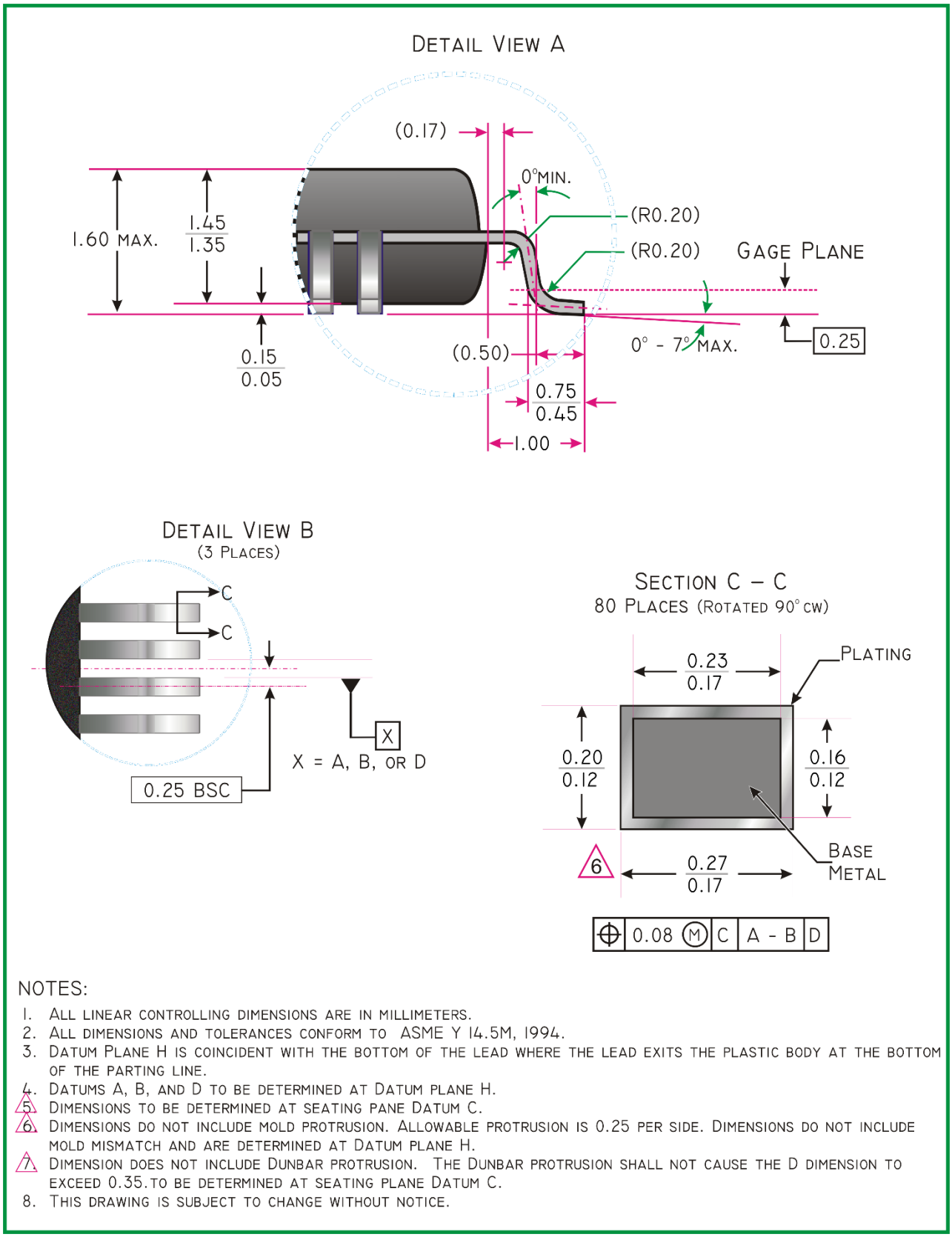


Figure 43 LQFP layout Detail Views

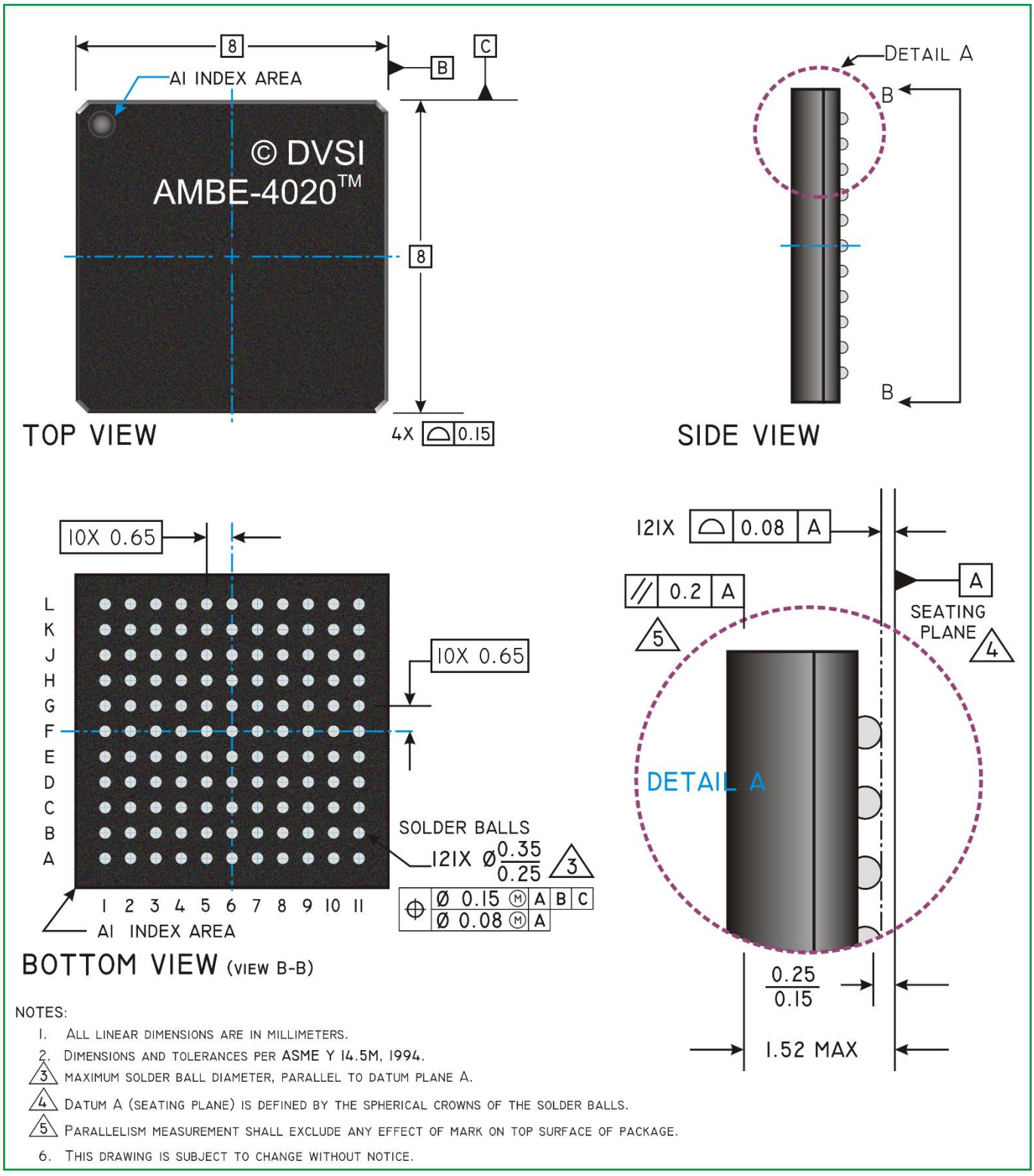


Figure 44 BGA Mechanical Details (Only available on Special Order)

8.2 Total Algorithmic Delay

The total delay due to the coding/decoding algorithm is shown below

Total Algorithmic Delay = 62 ms

8.3 Vocoder Rate by Index Number

Vocoder Rates by Index Number				
AMBE-1000™ Compatible Rates				
Rate Index #	**Total Rate**		**Speech Rate**	**FEC Rate**
0	2400		2400	0
1	3600		3600	0
2	4800		3600	1200
3	4800		4800	0
4	9600		9600	0
5	2400		2350	50
6	9600		4850	4750
7	4800		4550	250
8	4800		3100	1700
9	7200		4400	2800
10	6400		4150	2250
11	3600		3350	250
12	8000		7750	250
13	8000		4650	3350
14	4000		3750	250
15	4000		4000	0
AMBE-2000™ Compatible Rates				
Rate Index #	**Total Rate**		**Speech Rate**	**FEC Rate**
16	3600		3600	0
17	4000		4000	0
18	4800		4800	0
19	6400		6400	0
20	8000		8000	0
21	9600		9600	0
22	4000		2400	1600
23	4800		3600	1200
24	4800		4000	800
25	4800		2400	2400
26	6400		4000	2400
27	7200		4400	2800
28	8000		4000	4000
29	9600		2400	7200
30	9600		3600	6000
31	2000		2000	0
32	6400		3600	2800
33	Not available			
34	Not available			
35	Not available			

36	Not available			
37	Not available			
AMBE-3000™ Compatible Rates				
Rate Index #	**Total Rate**		**Speech Rate**	**FEC Rate**
38	3000		3000	0
39	3600		3600	0
40	4000		4000	0
41	4400		4400	0
42	4800		4800	0
43	6400		6400	0
44	7200		7200	0
45	8000		8000	0
46	9600		9600	0
47	Not available			
48	3600		3350	250
49	4000		3750	250
50	4800		4550	250
51	Not available			
52	Not available			
53	Not available			
54	Not available			
55	Not available			
56	4800		3600	1200
57	4800		4000	800
58	6400		4000	2400
59	7200		4400	2800
60	8000		4000	4000
61	9600		3600	6000
62	Not available			
63*	3600		2400	1200

Table 71 Rate Index Numbers

Notes
Rate Index #32 is compatible with the AMBE-2000™ Vocoder chip however; it is not part of the AMBE-2000™ Vocoder chip standard rate table.
Index rates #32 to #63 are AMBE+2 mode rates
***Index Rate #63 - This rate is interoperable with DSTAR**

Table Key for Table 71 Rate Index Numbers	
AMBE-1000™ Rates (AMBE™ Vocoder)	
AMBE-2000™ Rates (AMBE+™ Vocoder)	
AMBE-3000™ Vocoder Chip Rates (AMBE+2™ Vocoder)	

8.4 Schematics

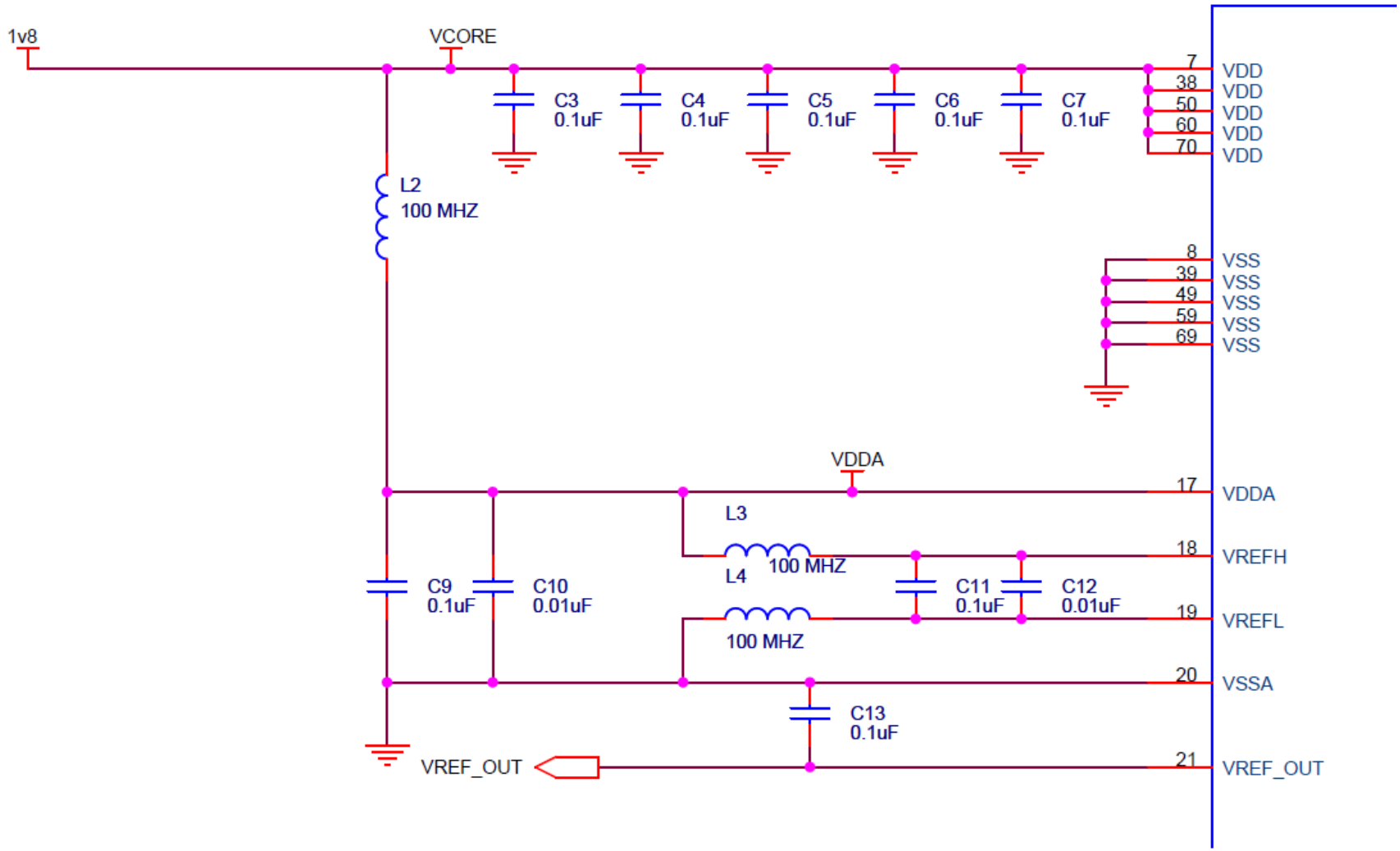


Figure 45 Single Supply Design

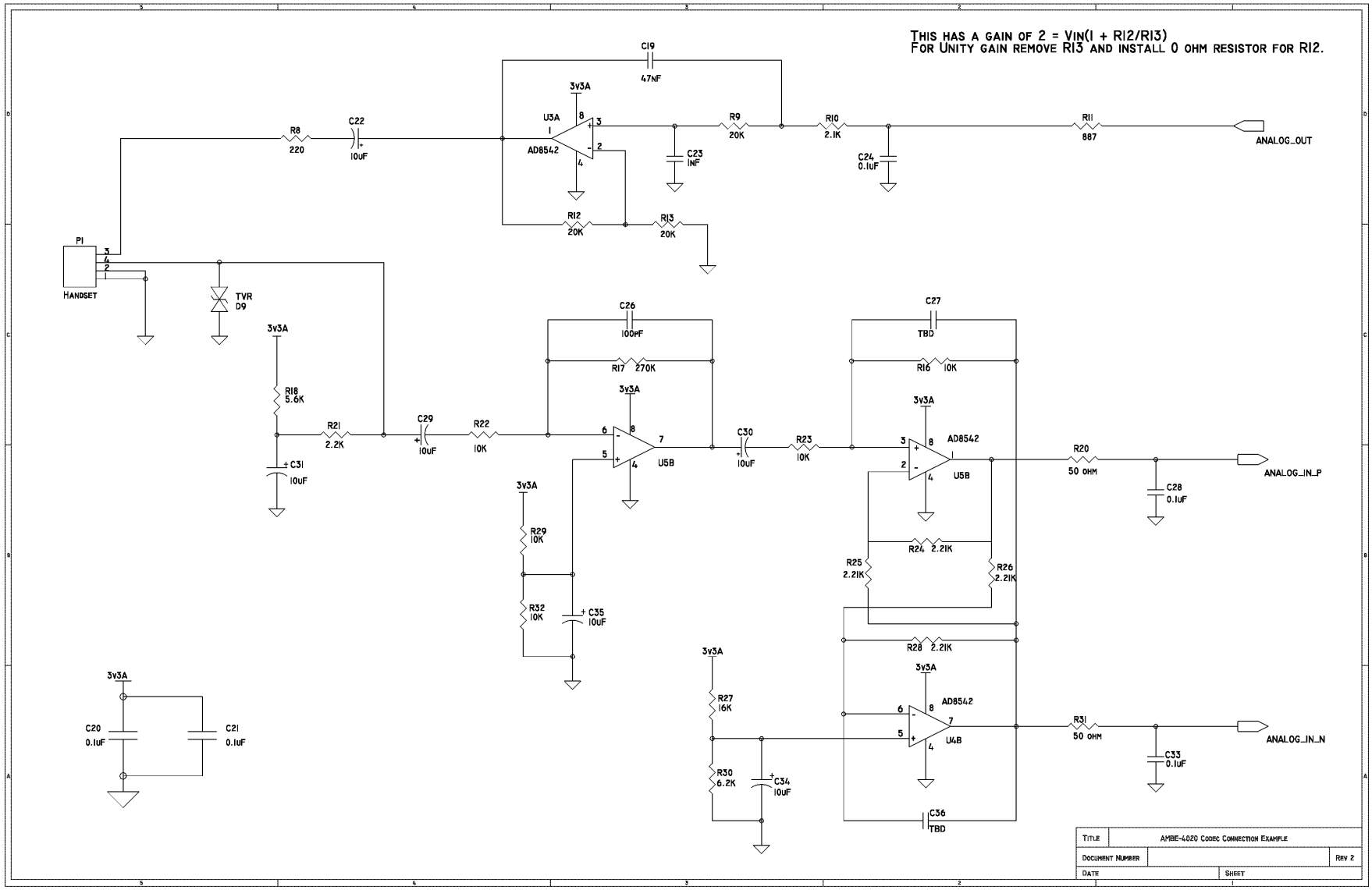


Figure 46 AMBE-4020™ Codec connection example

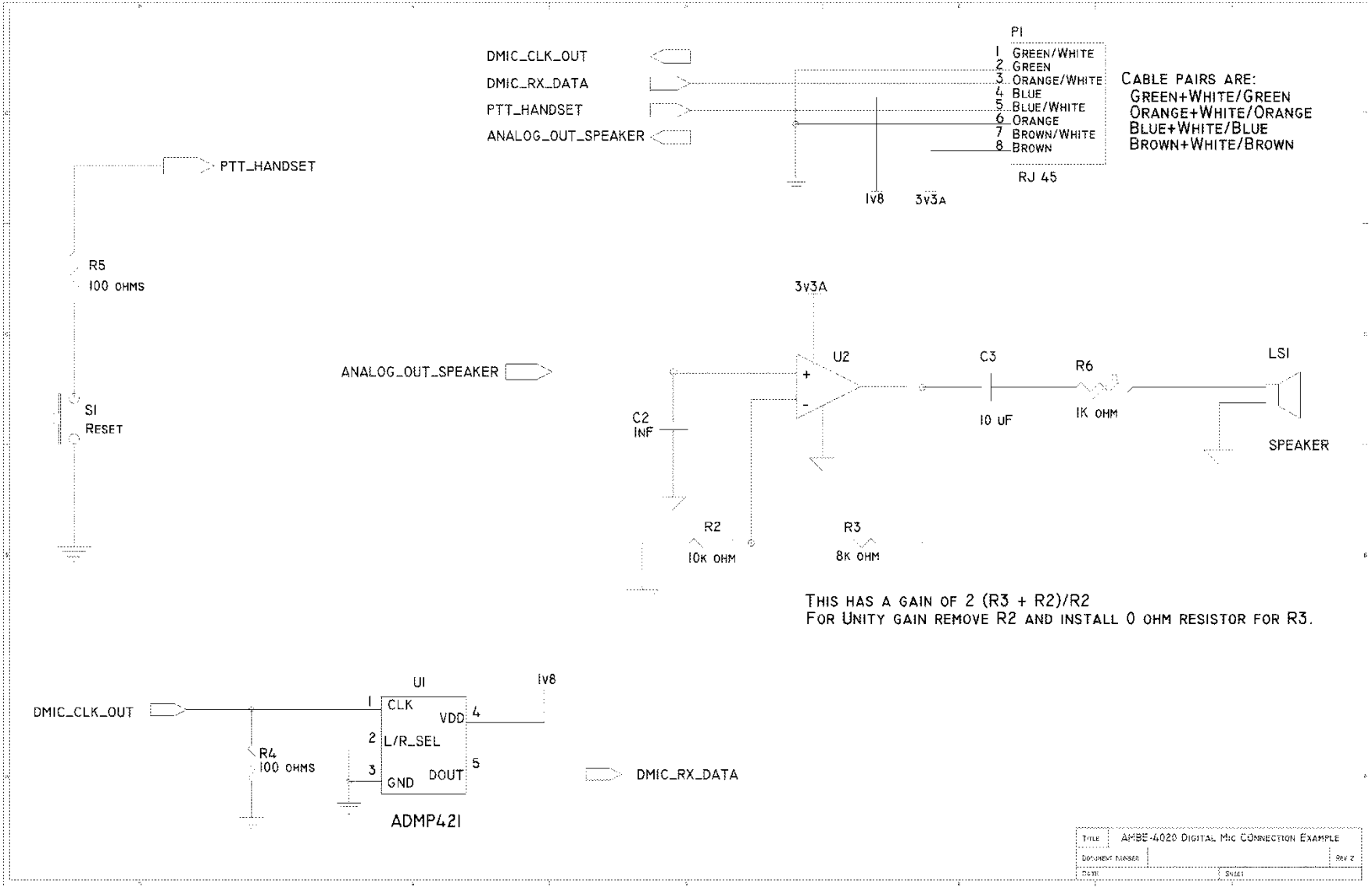


Figure 47 AMBE-4020™ DMIC connection example

SECTION 9

9 Support

9.1 DVSI Contact Information

If you have questions regarding the AMBE-4020™- Vocoder Chip please contact:

Digital Voice Systems, Inc.
600 Cleveland Street
STE 795
Clearwater, FL 33755 USA

Phone: (978) 392-0002
Fax: (978) 392-8866

email: mailto: techsupport@dvsinc.com
web site: http://www.dvsinc.com/

SECTION 10

10 Environmental and Compliance Information

Environmental specifications and compliances can be found on Freescale's website. Search for information regarding the MK10DX128VLK7 chip.

DVSI Part Number	AMBE-4020™ Vocoder Chip
Freescale Part Number	MK10DX128VLK7
Pb-Free	Yes
RoHS Compliant	Yes
Halogen Free	Yes
RoHS Certificate of Analysis (CoA)	Download RoHS CoA Report from http://www.freescale.com/
2nd Level Interconnect	e3
Moisture Sensitivity Level (MSL)	3
Floor Life	168 HOURS
Peak Package Body Temperature (PPT)	260°C
Maximum Time at Peak Temperature (s)	40
Number of Reflow Cycles	3
REACH SVHC	REACH Statement from http://www.freescale.com/

SECTION

11 IC Chip Software Errata

11

DVSI reserves the right to make modifications, enhancements, improvements and other changes to the AMBE-4020™ Vocoder Chip at any time without notice. This errata section provides up-to-date information regarding software developments as it pertains to the release number and release date. To identify the software release number of the AMBE-4020™ Vocoder Chip refer to the PKT_VERSTRING field in Section Packet Fields.

Release 00100 08-14-14
Original AMBE-4020™ release August 2014

12 History of Revisions

SECTION

12

History of Revisions			
Revision Number	Revision Date	Description	Pages
0.9	August, 2014	Preliminary Release	
1.0	January 2015	Edited Figure 5 AMBE-4020™ Vocoder Chip Pins for BGA Package Pins G1 and G2 were reversed	3
		Edited Table 2 Pinout List VDD BGA pins	
1.1	May 2015	Edited Figure 29 Basic Operation	46
		Removed Preliminary Status	10
1.2	July 2015	Removed crystal external capacitor specs. Added note for C_{X1} and C_{X2}.	23
2.0	December 2015	Added in details of the AMBE-4020 Full-Duplex Vocoder Chip	various
		Removed the L1 Inductor (100MHz) from the single supply design Figure 45 Single Supply Design.	111
2.1	March 2016	Added in Full Duplex Marking information. Notice regarding BGA package only available as special order.	Cover, 3, 4, 6, 108
2.2	November 2017	Corrected Typo in Table 2.5 and Figure 2.4. CODEC_RX_CLK signal on the BGA chip (pin C7) was incorrectly identified as pin D6.	6, 7
2.3	September 2018	**END USER PRODUCT License Agreement**	
2.4	June 2019	Edited no connection list in Table 2 Pinout List	10
2.5	Dec 2020	Edited typo in PKT_GAIN range it is +20 dB to -20 db (not ±90)	90
2.6	October 2021	**END USER PRODUCT License Agreement**	