ISD61S00 ChipCorder Telephony Feature Chip Design Guide

©2008 Nuvoton Technology Corporation America all rights reserved.

Important Notice

Nuvoton products are not designed, intended, authorized or warranted for use as components in systems or equipment intended for surgical implantation, atomic energy control instruments, airplane or spaceship instruments, transportation instruments, traffic signal instruments, combustion control instruments, or for other applications intended to support or sustain life. Further more, Nuvoton products are not intended for applications wherein failure of Nuvoton products could result or lead to a situation wherein personal injury, death or severe property or environmental damage could occur.

Nuvoton customers using or selling these products for use in such applications do so at their own risk and agree to fully indemnify Nuvoton for any damages resulting from such improper use or sales.

Table of Contents

1. GENERAL DESCRIPTION 10
2. FEATURES 10
3. PIN CONFIGURATION 12
4. PIN DESCRIPTION 13
5. BLOCK DIAGRAM 16
6. CONFIGURATION REGISTER MAP 17
6.1 Configuration Register Groups 17
6.2 Register Map 17
7. DEVICE STATUS 32
7.1 Device Status Register 32
8. FUNCTIONAL DESCRIPTION 34
8.1 SPI Interface 34
8.2 Record and Playback Control 36
8.2.1 COMP_CFG – Compression Configuration 36
8.2.2 COMP_CTRL – Compression Control 38
8.2.3 COMP_SRC – Compression Source 39
8.2.4 CLK_CTRL – Clock Control 39
8.2.5 CFG17 – Update Mode 40
8.3 Clock Generation and PLL 40
8.3.1 Phase Locked Loop (PLL) Design Example 41
8.3.2 PLL_CLK 42
8.3.3 PLL_FRAC 43
8.3.4 PLL_CTRL 43
8.4 Digital Audio Interface Control 43
8.4.1 PCM_CFG 44
8.4.2 PCM_TSLOTL 45
8.4.3 PCM_TSLOTR 45
8.4.4 PCM_COMP0 45
8.4.5 PCM_TSLOTL1 45
8.4.6 PCM_TSLOTR1 46
8.4.7 PCM_COMP1 46
8.4.8 PCM_TX_SEL 47
8.4.9 PCM_RX_SEL 48
8.5 GPIO and Interrupt Configuration 48
8.5.1 PORT_CFG 48
8.5.2 PORTx_IE 49
8.5.3 IE0 49
8.5.4 IE1 50

8.5.5 PORTA_DO 50
8.5.6 PORTA_OE 50
8.5.7 PORTA_PE 50
8.5.8 PORTA_DI 51
8.5.9 PORTA_PS 51
8.5.10 PORTA_DS 51
8.5.11 PORTA_AF 51
8.6 16 bit Indirect Index register R0-R7 53
8.7 Microphone interface and Auxiliary Interface 54
8.7.1 MIC_CTRL 54
8.7.2 MIC_BIAS 55
8.8 PSTN Analog Input 60
8.8.1 TI_GAIN 61
8.8.2 TI_CTRL 61
8.9 Analog Outputs 63
8.9.1 ANA_OUT 64
8.9.2 ANA_CTRL 64
8.10 DTMF Detection 65
8.10.1 DTMF_CTRL 67
8.10.2 DTMF_FIFO_CTRL 67
8.10.3 DTMF_FIFO 68
8.10.4 DTMF_FIFO_STATUS 68
8.10.5 DTMF_THRES 68
8.10.6 DTMF_PDT 69
8.10.7 DTMF_ADT 70
8.10.8 DTMF_ACCT 70
8.10.9 70
8.10.10 DTMF_RX_DATA 70
8.10.11 DTMF_ROW_FREQ 71
8.10.12 DTMF_COL_FREQ 71
8.10.13 Tip & Tricks 71
8.11 DTMF and Arbitrary Tone Generation. 72
8.11.1 TONE_CTRL 73
8.11.2 TONE_FREQ_A 73
8.11.3 TONE_FREQ_B 74
8.11.4 TONE_LEVEL_A 74
8.11.5 TONE_LEVEL_B 74
8.11.6 TONE_ON_TIME 74
8.11.7 TONE_OFF_TIME 75
8.11.8 TONE_LENGTH 75
8.11.9 75
8.11.10 TONE_INDEX_0 ~ TONE_INDEX_A 76
8.12 FSK Generation 76
8.12.1 FSKE_CTRL1 78

8.12.2 FSKE_CTRL2 78
8.12.3 FSKE_TX_DATA 79
8.12.4 FSKE_STATUS 79
8.12.5 FSKE_GAIN 79
8.12.6 FSKE_PROG 79
8.12.7 FSKE_BAUD (SW_MODE = 1) 80
8.12.8 FSKE_MARK_FREQ (SW_MODE = 1) 80
8.12.9 FSKE_SPACE_FREQ (SW_MODE = 1) 81
8.12.10 FSKE_TEST 81
8.12.11 81
8.12.12 Example FSK Generator Usage 81
8.13 FSK Detection 85
8.13.1 FSKD_CTRL 86
8.13.2 FSKD_MODE 88
8.13.3 FSKD_ADJUST 88
8.13.4 FSKD_STATUS 89
8.13.5 FSKD_THRES 89
8.13.6 FSKD_JUDGE 90
8.13.7 FSKD_SYNC 90
8.13.8 FSKD_CDET 90
8.13.9 FSKD_FIFO_CTRL 91
8.13.10 FSKD_FIFO_DOUT 92
8.13.11 FSKD_FIFO_STATUS 92
8.13.12 FSKD_ENERGY_HIGH_TH 92
8.13.13 FSKD_ENERGY_LOW_TH 93
8.13.14 FSKD_ENERGY_TC 93
8.13.15 93
8.13.16 FSKD_CDB_FREQ_LOW_CNT 93
8.13.17 FSKD_CDB_FREQ_HIGH_CNT 94
8.13.18 FSKD_LIMIT_TH 94
8.13.19 Example FSK Detector Usage 95
8.14 CAS and Arbitrary Tone (ATD) Detector 99
8.14.1 CAS_CTRL 101
8.14.2 CAS_THRES_LOW – CAS detector low threshold 102
8.14.3 CAS_MULT 102
8.14.4 CAS_PRESENT 103
8.14.5 CAS_ABSENT 104
8.14.6 CAS_STATUS 104
8.14.7 CAS_MODE - Arbitrary Tone Detection Mode 104
8.14.8 ATD_MAX_HFC (ATD_MODE = 1) 105
8.14.9 ATD_MIN_HFC (ATD_MODE = 1) 105
8.14.10 ATD_MAX_LFC (ATD_MODE = 1) 106
8.14.11 ATD_MIN_LFC (ATD_MODE = 1) 106
8.14.12 External Dual-IIR Coefficient (ATD_MODE= 1) 106

8.14.13 FSK_COEFF – FSK Encoder Coefficient RAM Data 107
8.14.14 FSK_COEFF_ADDR - FSK Encoder Coefficient RAM Address 108
8.14.15 CAS_MAX_HFC (FINE_TUNE = 1) 108
8.14.16 CAS_MIN_HFC (FINE_TUNE = 1) 109
8.14.17 CAS_MAX_LFC (FINE_TUNE = 1) 109
8.14.18 CAS_MIN_LFC (FINE_TUNE = 1) 109
8.14.19 110
8.14.20 110
8.14.21 Arbitrary Tone Detector Example 110
8.15 Voice Energy Detection (Speech Energy Detection) 115
8.15.1 VD_CTRL 116
8.15.2 VD_STATUS 116
8.15.3 VD_THRES 116
8.15.4 VD_ENERGY 116
8.16 Call Progress Tone Detector 117
8.16.1 CPT_CTRL 117
8.16.2 CPT_STATUS 118
8.16.3 CPT_THRES_H 118
8.16.4 CPT_THRES_L 118
8.16.5 CPT_ENERGY 119
8.17 Ring Detection and Pulse / Period Width Measurement 119
8.17.1 RNG_CTRL 120
8.17.2 RNG_STATE 122
8.17.3 RNG_CNTR 122
8.17.4 RNG_LATCH 123
8.17.5 123
8.18 Timer 123
8.18.1 123
8.18.2 TIME_TARG 124
8.18.3 TIME_CNT 124
8.19 Gain Stage and Mixer 124
8.19.1 GS_CTRL 128
8.19.2 CODEC EC Gain Stages 129
8.19.3 AECin Path Mixing Gain Control 130
8.19.4 LECin Path Mixing Gain Control 130
8.19.5 REC Path Mixing Gain Control Registers 131
8.19.6 I2SLin Path Mixing Gain Control Registers 132
8.19.7 I2SRin Path Mixing Gain Control Registers 132
8.19.8 Mixer Source Enable Registers 133
8.20 Air and Line CODEC 133
8.20.1 AC_EN, LC_EN – Air /Line CODEC Enable Register 134
8.20.2 AC_CTRL, LC_CTRL – Air /Line CODEC Dither Control 135
8.20.3 AC_ADCG, LC_ADCG – Air/Line CODEC ADC Gain. 135
8.20.4 AC_DACG, LC_DACG – Air/Line CODEC DAC Gain. 135

8.20.5 136
8.20.6 136
8.21 RINGER TONE GENERATOR 136
8.21.1 PWM Clock 137
8.21.2 PWM Tone1 Control 137
8.21.3 PWM Tone1 Frequency 138
8.21.4 PWM Tone2 Control 138
8.21.5 PWM Tone2 Frequency 139
9. ACOUSTIC PROCESSING BLOCK 139
9.1 Full/Half AEC Block Diagram 139
9.2 Control Register Memory Map 141
9.2.1 Threshold and Power Calculation 145
9.3 Control Registers 145
9.3.1 CONFIG 146
9.3.2 RESET 146
9.3.3 EC_BELTA 147
9.3.4 AS_COEFF 148
9.4 Double Talk Detector Control Registers 148
9.4.1 Function 149
9.4.2 DT_LONG_TC 149
9.4.3 DT_SHORT_TC 149
9.4.4 Double Talk Detector Parameters 150
9.4.5 DIVERGENCE 150
9.5 Voice Detector Control Registers 151
9.5.1 Function 151
9.5.2 VD_LONG_TC 151
9.5.3 VD_SHORT_TC 151
9.5.4 Voice Detector Parameters 151
VD_AVE_TC 152
9.5.5 VD_AVE_THRESH 153
9.6 AS1 & AS2 Control Registers 153
9.6.1 Function 153
9.6.2 AS1_BUILD_UP_TIME 153
9.6.3 AS1_MAX_ATTEN 153
9.6.4 AS2_BUILD_UP_TIME 154
9.6.5 AS2_MAX_ATTEN 154
9.7 Noise Suppressor Registers 154
9.7.1 Function 154
9.7.2 NS_POWER_ATTACK_TC 155
9.7.3 NS_ATTEN _TC 155
9.7.4 NS_ACTIVE_THRESHOLD 155
9.8 155
9.9 Soft Clip (SC) Control Registers 156

9.9.1 Functional Description 156
9.9.2 SC_CTRL 156
9.9.3 SC_NORMAL_INDEX 157
9.9.4 SC_LOW_INDEX 158
9.9.5 SC_THRESH 158
9.9.6 SC_POWER_ATTACK_TC 158
9.9.7 SC_GAIN_TC 159
9.10 State Read back Registers 159
9.10.1 Functional Description 159
9.10.2 Power Monitor 159
9.10.3 Signal Monitor 160
9.11 160
9.12 Automatic Gain Control 161
9.12.1 AGC_CTRL 162
9.12.2 AGC_INIT_GAIN 163
9.12.3 AGC_GAIN_HOLD 163
9.12.4 AGC_INC_DEC 163
9.12.5 AGC_ATK_DCY 163
9.12.6 AGC_GAIN_READ 164
9.12.7 AGC_STATE 164
9.12.8 AGC_PWR_TC 164
9.12.9 AGC_PK_TC 165
9.12.10 AGC_PK 165
9.12.11 AGC TARGETS 165
9.12.12 AGC NOISE PARAMETERS 166
10. SPI COMMANDS 166
10.1 Audio Play and Record Commands 169
10.1.1 Play Voice Prompt 170
10.1.2 Play Voice Prompt @Rn, n = 0 ~ 7 170
10.1.3 Play Voice Prompt, Loop 170
10.1.4 Play Voice Prompt, Loop, @Rn, n = 0 ~ 7 172
10.1.5 Stop Loop-Play Command 172
10.1.6 Execute Voice Macro 172
10.1.7 Execute Voice Macro @Rn, n = 0 ~ 7 173
10.1.8 Record Message 173
10.1.9 Record Message at Address 174
10.1.10 Play Message at Address 174
10.1.11 Play Silence 175
10.1.12 Stop Command 175
10.1.13 Erase Message at Address 176
10.1.14 SPI Send Audio Data 176
10.1.15 SPI Receive Audio Data 177
10.1.16 SPI Send Compressed Audio Data for direct programming to flash 179

10.1.17 SPI Receive previous REC_MSG or REC_MSG@ compressed Audio Data stored in flash 180
10.1.18 SPI Send Compressed Data to Decode 181
10.1.19 SPI Receive Encoded Data 182
10.2 Device Status Commands. 183
10.2.1 Read Status 183
10.2.2 Read Interrupt 183
10.2.3 Read Recorded Message Address Details 183
10.2.4 Read Message Length 184
10.2.5 Read ISD61S00 ID 184
10.3 Digital Memory Commands. 185
10.3.1 Digital Read 185
10.3.2 Digital Write 185
10.3.3 Erase Memory 186
10.3.4 Chip Erase 186
10.3.5 CHECKSUM 187
10.4 Device Configuration Commands 187
10.4.1 PWR_UP – Power up 188
10.4.2 PWR_DN – Power Down 188
10.4.3 WR_CFG_REG – Write Configuration Register 188
10.4.4 RD_CFG_REG – Read Configuration Register 188
10.5 Device Power Up Sequence 189
11. ISD61S00 MEMORY MANAGEMENT 189
11.1 ISD61S00 Memory Format 189
11.2 Message Management 190
11.2.1 Voice Prompts 191
11.2.2 Voice Macros 191
11.2.3 User Data 192
11.2.4 Reserved Sectors 192
11.2.5 Message Recordings 192
11.3 Memory and Message Headers 193
11.3.1 Memory Header 193
11.3.2 Message Header 194
11.4 Digital Access of Memory 194
11.5 Device Erase Commands 195
11.6 Memory Contents Protection 195
12. DEVICE INITIALIZATION 195
13. APPLICATION REFERENCE SCHEMATICS 196
13.1 PCB Layout Guidelines 196
14. PACKAGE SPECIFICATION 199
14.1 LQFP48L (7x7x1.4mm footprint 2.0mm) 199
15. ELECTRICAL CHARACTERISTICS 200
15.1 Absolute Maximum Ratings 200

15.2 Operating Conditions 200
15.3 DC Parameters 201
15.4 Analog Transmission Characteristics 202
15.5 Analog Distortion and Noise Parameters 202
15.5.1 8kHz sampling 202
15.5.2 16kHz sampling 203
15.6 SPI Timing 203
15.7 Recommended Clock/Crystal Specification 204
15.8 Dual Tone Alert Signal (CAS) 205
15.9 FSK Detection – 1200baud Bell 202, ITU V.23, 300 baud Bell 103, ITU V.21 206
15.10 FSK Transmitter – Bell 202, ITU-V.23, Bell 103, ITU-V.21 208
15.11 DTMF Detection 209
16. ORDERING INFORMATION 210
17. REVISION HISTORY 211

新唐科技 NUVOTON INTELLECTUAL PROPERTY

1. GENERAL DESCRIPTION

The ISD61S00 is a feature chip for the security and telephony industry. The device incorporates audio storage with a powerful macro scripting ability to facilitate audio prompting in a multi-language environment and simple address-free recording and playback. It utilizes external serial flash memory for audio data storage. In addition, the device includes circuitry to perform telephony based data communications including DTMF detection and generation, FSK modem functions from 75-1200 baud, CAS and CPT (Call Progress Tone) detection and ring detection. The audio path of the device is designed to interface to both the air side and line side of a PSTN system. The air side includes a flexible microphone interface incorporating a bias generator and gain control and a differential analog output for driving a speaker or power amplifier. The line side incorporates a line driver to drive PSTN loads and two variable gain differential inputs. The audio path includes full and half-duplex acoustic and line echo cancellation to implement full and half-duplex speakerphone functions. The digital audio interface can be configured to I^2S or PCM mode for digital serial audio communication. Control, monitoring and device programming is perfomed via a SPI interface. The device package is LQFP-48L.

2. FEATURES

- External Memory:
  - The ISD61S00 supports the following flash:

Manufacturer	Winbond		Numonyx			MXIC
Family	25X	25Q	25P	25PX	25PE	25L / 25V
JEDEC ID	EF 30 1X	EF 40 1X	20 20 1X	20 71 1X	20 80 1X	C2 20 1X

  - The addressing ability of ISD61S00 is up to 128Mbit, which is 64-minute record/playback time based on 8kHz/4bit ADPCM.
- Fast pre-recording: Recording is limited by the write rate of the attached external flash.
- Operating voltage: 2.7-3.6V.
- Sampling frequency: Recording and playback sampling frequencies of 4, 5.3, 6.4, 8, 10.6, 12.8 and 16 kHz.
- Compression algorithm:
  - For record and playback:
  - ADPCM compression at 2, 3, 4 or 5 bits per sample.
  - µ-Law companding at 6, 7 or 8 bits per sample.
  - Differential µ-Law encoding at 6, 7 or 8 bits per sample.
  - PCM encoding at 8, 10 or 12 bits per sample.
- For pre-recorded audio playback (Voice prompts) additionally:
  - Enhanced ADPCM compression at 2, 3, 4 or 5 bits per sample.
  - Multi-Bit rate optimized compression. This allows best possible compression given a metric of SNR and background noise levels.
- Message Management:
  - Simple address free real-time recording.
  - Flexible Voice Prompt message management and Voice Macro scripting for pre-recorded messages.
- External serial flash memory interface: Support up to 128Mbit for audio and digital data storage, equivalent to 64 minutes based on 8khz 4bit ADPCM.
- Digital access to flash memory: Memory can be reserved on a 4Kbyte sector basis for use as

conventional digital memory by host.

- Telephony/Modem/Data Features at 8KHz CODEC sampling rate:
  - DTMF encoder with 20 digit dial string buffer.
  - DTMF detector.
  - FSK generation at 75/110/150/300/1200 baud for Bell 103, Bell 202, V.21 or V.23 modem standards. Transmit FIFO to reduce host interaction.
  - FSK detector at 75/110/150/300/1200 baud for Bell 103, Bell 202, V.21 or V.23 modem standards with receive FIFO.
  - Ring Detector.
  - Call Progress Tone (CPT) detector for detecting dial tones, busy tones etc.
  - CAS detector for caller ID type I, type II implementation.
  - Arbitrary dual tone generator.
  - Arbitrary tone detector.
- Acoustic Echo Cancellation (AEC), Line Echo Cancellation (LEC) and Automatic Gain Control (AGC) for on-chip speakerphone support.
- Up to 17 GPIO pins accessible through the SPI interface.
- Audio Input:
  - TI1 and TI2: Differential Analog inputs for PSTN interface (on-hook and off-hook).
  - MIC+/-: Analog interface to microphone.
- Audio Output:
  - PO: Differential Analog output for PSTN interface.
  - SPK: Differential output buffer for speaker driver.
- I/O:
  - SPI interface: MISO, MOSI, SCLK, SS for commands and digital audio data.
  - INT and R/B signal for signaling and flow control.
- Package: Green LQFP-48L
- Temperature: -40°C to 85°C

新唐科技 NUVOTON INTELLECTUAL PROPERTY

3. PIN CONFIGURATION

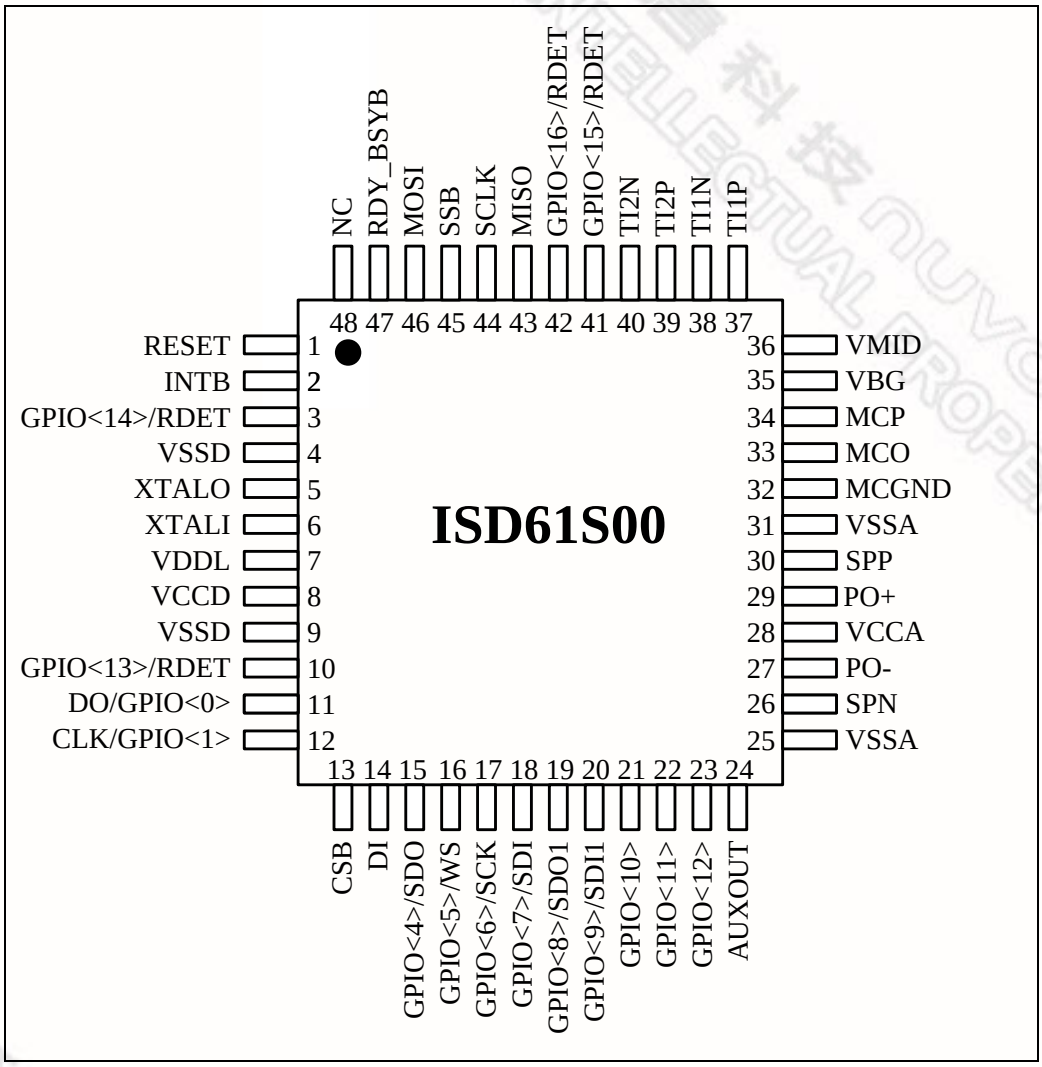


Figure 3-1 ISD61S00 48-Lead LQFP Pin Configuration.

4. PIN DESCRIPTION

Pin Number	Pin Name	I/O	Function	Drive
1	RESET	I	Raising this pin to VCC will reset the chip.	N/A
2	INTB	O	Active low interrupt request pin. This pin has an open drain output.	4/8mA
3	GPIO<14>/RDET	I/O*	General Purpose IO Pin or the Ring Detect input (RDET).	4/8mA
4	VSSD	G	Digital Ground.	N/A
5	XTALOUT	O	Crystal Interface output pin.	N/A
6	XTALIN	I	Crystal interface input pin. It can also be used to provide an external clock to the device.	N/A
7	VDDL	O	This pin has a nominal 1.8V output to supply the internal logic. A 2.2nF capacitor should be connected to this pin.	N/A
8	VCCD	P	Digital power supply pin	N/A
9	VSSD	G	Digital Ground.	N/A
10	GPIO<13>/RDET	I/O*	General Purpose IO Pin.	4/8mA
11	DO/GPIO<0>	I/O	Flash interface data out. Alternatively General Purpose IO Pin.	4/8mA
12	CLK/GPIO<1>	I/O	Flash interface clock. Alternatively General Purpose IO Pin.	4/8mA
13	CSB	O	Flash interface chip select bar.	4/8mA
14	DI	I	Flash interface data in.	4/8mA
15	GPIO<4>/SDO	I/O*	General Purpose IO Pin. Serial Data Out for the I2S interface.	8/16mA
16	GPIO<5>/WS	I/O*	General Purpose IO Pin. Word Select (WS) output for the I2S Interface	8/16mA
17	GPIO<6>/SCK	I/O*	General Purpose IO Pin. Serial Clock output for the I2S Interface	8/16mA
18	GPIO<7>/SDI	I/O*	General Purpose IO Pin. Serial Data Input (SDI) input pin for the I2S Interface.	8/16mA
19	GPIO<8>/SDO1	I/O*	General Purpose IO Pin. Secondary Serial Data Out for the I2S interface.	4/8mA
20	GPIO<9>SDI1	I/O*	General Purpose IO Pin. Secondary Serial Data Input (SDI) input pin for the I2S Interface.	4/8mA

21	GPIO<10>	I/O*	General Purpose IO Pin.	4/8mA
22	GPIO<11>	I/O*	General Purpose IO Pin or the Ring Detect input (RDET).	4/8mA
23	GPIO<12>	I/O*	General Purpose IO Pin.	4/8mA
24	AUXOUT	O	Auxiliary output from PSTN or SPEAKER DAC.	N/A
25	VSSA	G	Analog Ground Pin	N/A
26	SPKN	O	Negative speaker driver output	N/A
27	PO-	O	Negative line driver output	N/A
28	VCCA	P	Analog Power Pin	N/A
29	PO+	O	Positive line driver output	N/A
30	SPKP	O	Positive speaker driver output	N/A
31	VSSA	G	Analog Ground Pin	N/A
32	MCGND	I	Analog ground pin for MIC. This pin should be connected to a quiet VSSA and used as the return for microphones connected to MCP.	N/A
33	MCO	I/O	MIC feedback signal	N/A
34	MCP	I	Positive MIC signal input.	N/A
35	VBG	I/O	Voltage reference, a 100nF capacitor should be connected to this pin.	N/A
36	VMID	I/O	Voltage reference, a 4.7uF capacitor should be connected to this pin.	N/A
37	TI1P	I	PSTN Line #1 positive input	N/A
38	TI1N	I	PSTN Line #1 negative input	N/A
39	TI2P	I	PSTN Line #2 positive input	N/A
40	TI2N	I	PSTN Line #2 negative input	N/A
41	GPIO<15>/RDET	I/O*	General Purpose IO Pin. Can be configured as Ring Detect (RDET) input.	4/8mA
42	GPIO<16>/RDET	I/O*	General Purpose IO Pin. Can be configured as Ring Detect (RDET) input.	4/8mA
43	MISO	O	SPI slave serial data output from the ISD61S00 to the host. This pin is tri-stated when SSB=1.	4/8mA
44	SCLK	I	SPI serial Clock input to the ISD61S00 from the host.	N/A
45	SSB	I	Slave select input to the ISD61S00 from the host	N/A
46	MOSI	I	SPI slave serial data input to the ISD61S00 from the host.	4/8mA

47	RDY/BSYB	O	This pin is at VCCD when the chip is ready to accept a new command/data and is at VSSD when device is busy.	4/8mA
48	NC			

* Default state for digital I/O pins is input with internal pull-up [38Kohms to 83Kohms] to VCCD supply

新唐科技 NUVOTON INTELLECTUAL PROPERTY

5. BLOCK DIAGRAM

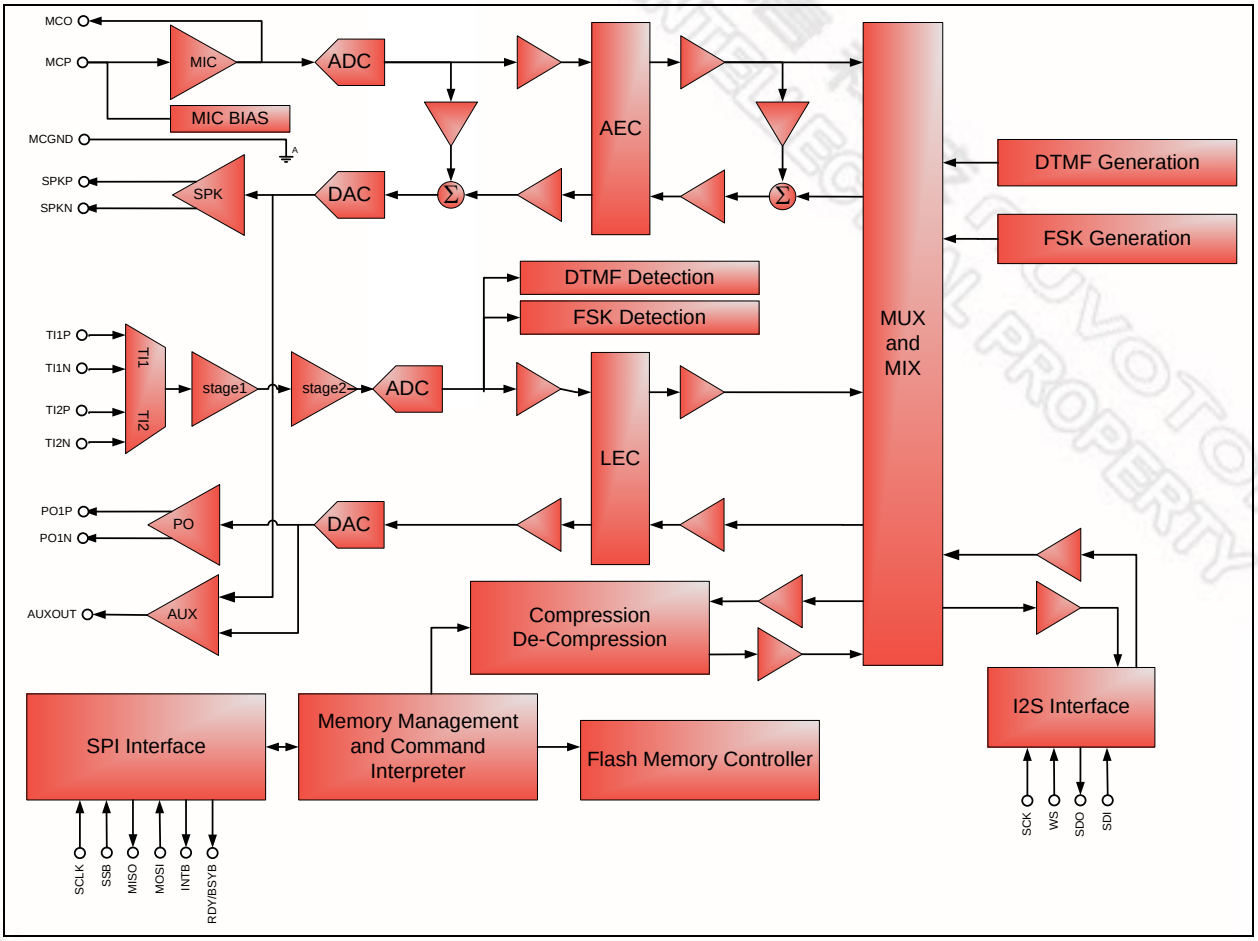


Figure 5-1 ISD61S00 Block Diagram

6. CONFIGURATION REGISTER MAP

6.1 Configuration Register Groups

Data memory address	Size (Byte)		Function
	Usage	ALL	
0x000 ~ 0x0FF		256	Device Control Registers
0x110 ~ 0x13F		32	Gain stage and Mixer Control Registers
0x140 ~ 0x14F		16	Air CODEC Control
0x150 ~ 0x15F		16	Line CODEC Control
0x160 ~ 0x17F		32	FSK Encoder Control
0x180 ~ 0x19F		32	FSK Decoder Control
0x1A0 ~ 0x1AF		16	Ringer Tone(PWM) Control Registers
0x1B0 ~ 0x1BF		16	Ring Detection PPM Control
0x1CO ~ 0x1DF		32	DTMF and Tone Generation
0x1E0 ~ 0x1EF		16	DTMF Detection Control
0x1F0 ~ 0x1FF		16	Voice Energy Detection Control
0x200 ~ 0x27F		128	CAS Detection Control
0x280 ~ 0x28F		16	CPT Detector Control
0x290 ~ 0x29F		16	Timer Control
0x300 ~ 0x3FF		256	AEC/LEC Control

6.2 Register Map

Addr.	Name	Mode	Value At Reset	Function	
0x000	AUD	R/W	0x64	Compression & SR ctrl	8.2.1
0x001	REG1	R/W	0x00	Compression Control	8.2.2
0x002	REG2	R/W	0x00	Compression Source	8.2.3
0x003	REG3	R/W	0x03	Clock Control	8.2.4
0x004	TI_GAIN	R/W	0x00	Input Amplifier Gain	8.8.1
0x005	TI_CTRL	R/W	0x00	Input Amplifier Control	8.8.2
0x006	MIC_CTRL	R/W	0x00	MIC Interface Control	8.7.1

Addr.	Name	Mode	Value At Reset	Function	
0x007	MIC_BIAS	R/W	0x00	MIC Interface Bias	8.7.2
0x008	ANA_OUT	R/W	0x00	Analog Output Control	8.9.1
0x009	ANA_CTRL	R/W	0x10	Analog Control	8.8.2
0x00A	ID_OVR	R/W	0x00	Memory Control	
0x00B	PORT_CFG	R/W	0x00	GPIO INT Ctrl	8.5.1
0x00C	PORTA_IE	R/W	0x00	Port A INT Enable	8.5.2
0x00D	PORTB_IE	R/W	0x00	Port B INT Enable	8.5.2
0x00E	PORTC_IE	R/W	0x00	Port C INT Enable	8.5.2
0x00F	DPLL Control	R/W	0x00	Digital PLL Control	8.5.2
0x014	IE0	R/W	0xFF	INT Enable 0	8.5.3
0x015	IE1	R/W	0xFF	INT Enable 1	8.5.4
0x017	UPDATE_MD	R/W	0x01	Update Ctrl	8.2.5
0x018	CHK_SUM_CTRL	R/W	0x01	Check sum control	
0x019	PORTA_DO	R/W	0x00	Port A Dout	8.5.5
0x01A	PORTA_OE	R/W	0x00	Port A Output Enable	8.5.6
0x01B	PORTA_PE	R/W	0xFF	Port A Pull Enable	8.5.7
0x01C	PORTA_DI	R	NA	Port A Din	8.5.8
0x01D	PORTA_PS	R/W	0xFF	Port A pull Select	8.5.9
0x01E	PORTA_DS	R/W	0x00	Port A Drive Strength	8.5.10
0x01F	PORTA_AF	R/W	0x03	Port A Alternate Function	8.5.11
0x020	R0_LSB	R/W	0x00	Indirect reg R0 LSB	8.6
0x021	R0	R/W	0x00	Indirect reg R0	8.6
0x022	R1_LSB	R/W	0x00	Indirect reg R1 LSB	8.6
0x023	R1	R/W	0x00	Indirect reg R1	8.6
0x024	R2_LSB	R/W	0x00	Indirect reg R2 LSB	8.6
0x025	R2	R/W	0x00	Indirect reg R2	8.6
0x026	R3_LSB	R/W	0x00	Indirect reg R3 LSB	8.6
0x027	R3	R/W	0x00	Indirect reg R3	8.6
0x028	R4_LSB	R/W	0x00	Indirect reg R4 LSB	8.6
0x029	R4	R/W	0x00	Indirect reg R4	8.6
0x02A	R5_LSB	R/W	0x00	Indirect reg R5 LSB	8.6
0x02B	R5	R/W	0x00	Indirect reg R5	8.6
0x02C	R6_LSB	R/W	0x00	Indirect reg R6 LSB	8.6
0x02D	R6	R/W	0x00	Indirect reg R6	8.6

Addr.	Name	Mode	Value At Reset	Function	
0x02E	R7_LSB	R/W	0x00	Indirect reg R7 LSB	8.6
0x02F	R7	R/W	0x00	Indirect reg R7	8.6
0x030	PORTB_DO	R/W	0x00	Port B Dout	8.5
0x031	PORTB_OE	R/W	0x00	Port B Output Enable	8.5
0x032	PORTB_PE	R/W	0xFF	Port B Pull Enable	8.5
0x033	PORTB_DI	R	NA	Port B Din	8.5
0x034	PORTB_PS	R/W	0xFF	Port B pull Select	8.5
0x035	PORTB_DS	R/W	0x00	Port B Drive Strength	8.5
0x036	PORTB_AF	R/W	0x00	Port B Alternate Function	8.5.11
0x037	PORTC_DO	R/W	0x00	Port C Dout	8.5
0x038	PORTC_OE	R/W	0x00	Port C Output Enable	8.5
0x039	PORTC_PE	R/W	0xFF	Port C Pull Enable	8.5
0x03A	PORTC_DI	R	NA	Port C Din	8.5
0x03B	PORTC_PS	R/W	0xFF	Port C pull Select	8.5
0x03C	PORTC_DS	R/W	0x00	Port C Drive Strength	8.5
0x03D	PORTC_AF	R/W	0xFC	Port C Alternate Function	8.5
0x040	PLL_CLK	R/W	0x20	PLL Clock Control	8.3.2
0x041	PLL_K2	R/W	0x00	PLL Fraction MSB	8.3.3
0x042	PLL_K1	R/W	0x00	PLL Fraction byte 1	8.3.3
0x043	PLL_K0	R/W	0x00	PLL Fraction LSB	8.3.3
0x044	PLL_CTRL	R/W	0x88	PLL Control Register	8.3.4
0x050	PCM_CFG	R/W	0x00	PCM Configuration	8.4.1
0x051	PCM_TSLOTL	R/W	0x01	Left Channel PCM MSB start	8.4.2
0x052	PCM_TSLOTR	R/W	0x11	Right Channel PCM MSB start	8.4.3
0x053	PCM_COMP0	R/W	0x00	PCM0 Compression	8.4.4
0x054	PCM_TSLOTL1	R/W	0x01	Left Channel PCM1 MSB start	8.4.5
0x055	PCM_TSLOTR1	R/W	0x11	Right Channel PCM1 MSB start	8.4.6
0x056	PCM_COMP1	R/W	0x00	PCM1 Compression	8.4.7
0x057	PCM_TX_SEL	R/W	0x00	PCM Transmission Select	8.4.8
0x058	PCM_RX_SEL	R/W	0x00	PCM Receive Select	8.4.9
0x059	CHKSUM SUM1 LSB	R	0x00	Chksum1[7:0]	
0x05A	CHKSUM SUM1	R	0x00	Chksum1[15:8]	
0x05B	CHKSUM SUM2 LSB	R	0x00	Chksum2[7:0]	
0x05C	CHKSUM SUM2	R	0x00	Chksum2[15:8]	

Addr.	Name	Mode	Value At Reset	Function	
0x05D	REC_MSG_LEN LSB	R/W	0x00	REC Message Sector Length [7:0]	
0x05E	REC_MSG_LEN	R/W	0x00	REC Message Sector Length [15:8]	
0x110	GS_CTRL	R/W	0x00	GS Control	
0x111	GS_ACST	R/W	0x40	Air CODEC Side tone	
0x112	GS_ACIG	R/W	0x00	Air CODEC input Gain	
0x113	GS_ACOG	R/W	0x00	Air CODEC output Gain	
0x114	GS_AEOG	R/W	0x00	AEC Output Gain	
0x115	GS_AEIG	R/W	0x00	AEC Input Gain	
0x116	GS_LCIG	R/W	0x00	Line CODEC input Gain	
0x117	GS_LCOG	R/W	0x00	Line CODEC output Gain	
0x118	GS_LEOG	R/W	0x00	LEC Output Gain	
0x119	GS_LEIG	R/W	0x00	LEC Input Gain	
0x11A	GS_AOAI	R/W	0x00	AEC out - AEC in Gain	
0x11B	GS_LOAI	R/W	0x00	LEC out - AEC in Gain	
0x11C	GS_PLAI	R/W	0x00	PLAY - AEC in Gain	
0x11D	GS_ILAI	R/W	0x00	I2SL - AEC in	
0x11E	GS_IRAI	R/W	0x00	I2SR - AEC in	
0x11F	GS_AOLI	R/W	0x00	AEC out - LEC in Gain	
0x120	GS_LOLI	R/W	0x00	LEC out - LEC in Gain	
0x121	GS_PLLI	R/W	0x00	PLAY - LEC in Gain	
0x122	GS_ILLI	R/W	0x00	I2SL - LEC in	
0x123	GS_IRLI	R/W	0x00	I2SR - LEC in	
0x124	GS_AORI	R/W	0x00	AEC out - REC in Gain	
0x125	GS_LORI	R/W	0x00	LEC out - REC in Gain	
0x126	GS_PLRI	R/W	0x00	PLAY - REC in Gain	
0x127	GS_ILRI	R/W	0x00	I2SL - REC in	
0x128	GS_IRRI	R/W	0x00	I2SR - REC in	
0x129	GS_AOIL	R/W	0x00	AEC out - I2SL in Gain	
0x12A	GS_LOIL	R/W	0x00	LEC out - I2SL in Gain	
0x12B	GS_PLIL	R/W	0x00	PLAY - I2SL in Gain	
0x12C	GS_ILIL	R/W	0x00	I2SL - I2SL in	
0x12D	GS_IRIL	R/W	0x00	I2SR - I2SL in	
0x12E	GS_AOIR	R/W	0x00	AEC out - I2SR in Gain	
0x12F	GS_LOIR	R/W	0x00	LEC out - I2SR in Gain	

NUVOTON CONFIDENTIAL PROPERTY

Addr.	Name	Mode	Value At Reset	Function	
0x130	GS_PLIR	R/W	0x00	PLAY - I2SR in Gain	
0x131	GS_ILIR	R/W	0x00	I2SL - I2SR in	
0x132	GS_IRIR	R/W	0x00	I2SR - I2SR in	
0x13B	MX_AEC	R/W	0x00	AEC Mix Ctrl	
0x13C	MX_LEC	R/W	0x00	LEC Mix Ctrl	
0x13D	MX_REC	R/W	0x00	REC Mix Ctrl	
0x13E	MX_ISL	R/W	0x00	I2S_L Mix Ctrl	
0x13F	MX_ISR	R/W	0x00	I2S_R Mix Ctrl	
0x140	AC_EN	R/W	0x20	Air CODEC Enable	
0x141	AC_CTRL	R/W	0x07	Air CODEC Control	
0x142	AC_ADCG_MSB	R/W	0x04	Air ADC Gain MSB	
0x143	AC_ADCG_LSB	R/W	0x00	Air ADC Gain LSB	
0x144	AC_DACG_MSB	R/W	0x04	Air DAC Gain MSB	
0x145	AC_DACG_LSB	R/W	0x00	Air DAC Gain LSB	
0x146	AC_ADOUT_MSB	R	NA	Air ADC Output MSB	
0x147	AC_ADOUT_LSB	R	NA	Air ADC Output LSB	
0x148	AC_DAOUT_MSB	R	NA	Air DAC Output MSB	
0x149	AC_DAOUT_LSB	R	NA	Air DAC Output MSB	
0x150	LC_EN	R/W	0x20	Line CODEC Enable	
0x151	LC_CTRL	R/W	0x07	Line CODEC Control	
0x152	LC_ADCG_MSB	R/W	0x04	Line ADC Gain MSB	
0x153	LC_ADCG_LSB	R/W	0x00	Line ADC Gain LSB	
0x154	LC_DACG_MSB	R/W	0x04	Line DAC Gain MSB	
0x155	LC_DACG_LSB	R/W	0x00	Line DAC Gain LSB	
0x156	LC_ADOUT_MSB	R	0x00	Line ADC Dout	
0x157	LC_ADOUT_LSB	R	0x00	Line ADC Dout LSB	
0x158	LC_DAOUT_MSB	R	0x00	Line DAC Dout MSB	
0x159	LC_DAOUT_LSB	R	0x00	Line DAC Dout LSB	
0x160	FSKE_CTRL1	R/W	0x00	FSK Encoder Control	8.12.1
0x161	FSKE_CTRL2	R/W	0x23	FSK Encoder Control	8.12.2

NUVOTON CONFIDENTIAL PROPERTY

Addr.	Name	Mode	Value At Reset	Function	
0x162	FSKE_FIFO_W	V	NA	FSK Encoder FIFO input	8.12.3
0x163	FSKE_STATUS	R	0x20	FSK Encoder Status	8.12.4
0x164	FSKE_GAIN	R/W	0x39	FSK Encoder Signal Gain	8.12.5
0x165	FSKE_PROG	R/W	0x00	FSK Encoder Programmable mode	8.12.6
0x166	FSKE_BAUD_2	R/W	0x00	FSK Encoder Baud Rate	8.12.7
0x167	FSKE_BAUD_1	R/W	0x00	FSK Encoder Baud Rate	8.12.7
0x168	FSKE_BAUD_0	R/W	0x00	FSK Encoder Baud Rate	8.12.7
0x169	FSKE_MARK_1	R/W	0x00	FSK Encoder Mark Freq	8.12.8
0x16A	FSKE_MARK_0	R/W	0x00	FSK Encoder Mark Freq	8.12.8
0x16B	FSKE_SPACE_1	R/W	0x00	FSK Encoder Space Freq	8.12.9
0x16C	FSKE_SPACE_0	R/W	0x00	FSK Encoder Space Freq	8.12.9
0x180	FSKD_CTRL	R/W	0x00	FSK Decoder Control	8.13.1
0x181	FSKD_MODE	R/W	0x04	FSK Decoder Mode	8.13.2
0x182	FSKD_ADJUST	R/W	0xC0	FSK Decoder Spec Adjustment	8.13.3
0x183	FSKD_STATUS	R	0x02	FSK Decoder Status	8.13.4
0x184	FSKD_THRES_MSB	R/W	0x00	FSK Decoder Hysteresis MSB	8.13.5
0x185	FSKD_THRES_LSB	R/W	0x80	FSK Decoder Hysteresis LSB	8.13.5
0x186	FSKD_JUDGE	R/W	0x04	FSK Decoder judge threshold	8.13.6
0x187	FSKD_SYNC	R/W	0x02	FSK Decoder Sync	8.13.7
0x188	FSKD_CDET	R/W	0x03	FSK Decoder Carrier Detect	8.13.8
0x189	FSKD_FIFO_CTRL	R/W	0x00	FSK Decoder FIFO control	8.13.9
0x18A	FSKD_FIFO_DOUT	V	NA	FSK Decoder FIFO Data	8.13.10
0x18B	FSKD_FIFO_STATUS	R	0x01	FSK Decoder FIFO Status	8.13.11
0x18C	FSKD_ENERGY_HI_TH	R/W	0x0C	FSK Decoder Carrier Energy Threshold	8.13.12
0x18D	FSKD_ENERGY_LO_TH	R/W	0x06	FSK Decoder Carrier Energy Threshold	8.13.13
0x18E	FSKD_ENERGY_TC	R/W	0x40	FSK Decoder Energy time constant	8.13.14
0x190	FSKD_CDB_FREQ_LOW_CNT	R/W	0x2F	FSK Carrier Detect Frequency Adjust	8.13.16
0x191	FSKD_CDB_FREQ_LOW_CNT_LSB	R/W	0x14	FSK Carrier Detect Frequency Adjust	8.13.16
0x192	FSKD_CDB_FREQ_HIGH_CNT	R/W	0x23	FSK Carrier Detect Frequency Adjust	8.13.17
0x193	FSKD_CDB_FREQ_HIGH_CNT_LSB	R/W	0xF5	FSK Carrier Detect Frequency Adjust	8.13.17
0x194	FSKD_LIMIT_TH	R/W	0xFF	FSK Signal Clamp Threshold	8.13.18

Addr.	Name	Mode	Value At Reset	Function	
0x1A0	PWM Clock	R/W	0x00	PWM Operation Clock Enable	8.21.1
0x1A2	PWM Tone1 Control	R/W	0x00	PWM Tone1 Control	8.21.2
0x1A3	PWM Tone1 Frequency	R/W	0x00	PWM Tone1 Frequency	8.21.3
0x1A4	PWM Tone2 Control	R/W	0x00	PWM Tone2 Control	8.21.4
0x1A5	PWM Tone2 Frequency	R/W	0x00	PWM Tone2 Frequency	8.21.5
0x1B0	RNG_CTRL	R/W	0x00	Ring Det. Control	8.17.1
0x1B1	RNG_STATE	R	0x00	Ring Det. State	8.17.2
0x1B2	RNG_CNTR_MSB	R	0x00	Ring Det. Counter	8.17.3
0x1B3	RNG_CNTR_LSB	R	0x00	Ring Det. Counter	8.17.3
0x1B4	RNG_LATCH	R/W	0x00	Ring Det. Counter latch control	8.17.4
0x1C0	TONE_CTRL	R/W	0x00	Tone Generator Control	8.11.1
0x1C1	TONE_FREQ_A_MSB	R/W	0x00	Tone Frequency A	8.11.2
0x1C2	TONE_FREQ_A_LSB	R/W	0x00	Tone Frequency A	8.11.2
0x1C3	TONE_FREQ_B_MSB	R/W	0x00	Tone Frequency B	8.11.3
0x1C4	TONE_FREQ_B_LSB	R/W	0x00	Tone Frequency B	8.11.3
0x1C5	TONE_LEVEL_A	R/W	0x00	Tone Level A	8.11.4
0x1C6	TONE_LEVEL_B	R/W	0x00	Tone Level B	8.11.5
0x1C7	TONE_ON_TIME	R/W	0x00	Present (on) time of each tone.	8.11.6
0x1C8	TONE_OFF_TIME	R/W	0x00	Absent (off) time of each tone.	8.11.7
0x1C9	TONE_BUF_LEN	R/W	0x00	The number of DTMF digits or tones to generate	8.11.8
0x1D0-0x1DB	TONE_BUFFER	R/W	0x00	Tone Buffer 0x1D0-0x1DB	8.11.10
0x1E0	DTMF_CTRL	R/W	0x00	DTMF Det. Control	8.10.1
0x1E1	DTMF_FIFO_CTRL	R/W	0x40	DTMF Det. FIFO Control	8.10.2
0x1E2	DTMF_FIFO	R	0x00	DTMF Det. FIFO Information	8.10.3
0x1E3	DTMF_FIFO_STATUS	R/W	0x01	DTMF Det. FIFO Status	8.10.4
0x1E4	DTMF_THRES_MSB	R/W	0x01	DTMF Det. Threshold MSB	8.10.5
0x1E5	DTMF_THRES_LSB	R/W	0x00	DTMF Det. Threshold LSB	8.10.5
0x1E6	DTMF_PDT	R/W	0x00	DTMF Det. Present Detect Time	8.10.6
0x1E7	DTMF_ADT	R/W	0x00	DTMF Det. Absent Detect Time	8.10.7
0x1E8	DTMF_ACCT	R/W	0x00	DTMF Det. Accept Time	8.10.8

Addr.	Name	Mode	Value At Reset	Function	
0x1EA	DTMF_RX_DATA	R	0x00	DTMF Det. Received Data	8.10.10
0x1EB	DTMF_RFREQ_MSB	R	0x00	DTMF Det. Row Frequency MSB	8.10.11
0x1EC	DTMF_RFREQ_LSB	R	0x00	DTMF Det. Row Frequency LSB	8.10.11
0x1ED	DTMF_CFREQ_MSB	R	0x00	DTMF Det. Column Frequency MSB	8.10.12
0x1EE	DTMF_CFREQ_LSB	R	0x00	DTMF Det. Column Frequency LSB	
0x1F0	VD_CTRL	R/W	0x00	Voice Det. Control	
0x1F1	VD_STATUS	R	0x00	Voice Det. Status	
0x1F2	VD_THRES_MSB	R/W	0x01	Voice Det. Threshold MSB	
0x1F3	VD_THRES_LSB	R/W	0x00	Voice Det. Threshold LSB	
0x1F4	VD_ENERGY_MSB	R	0x00	Voice Det. Energy MSB	
0x1F5	VD_ENERGY_LSB	R	0x00	Voice Det. Energy LSB	
0x200	CAS_CTRL	R/W	0x40	CAS Control	
0x201	CAS_THRES_LOW	R/W	0x00	CAS Low Threshold	
0x202	CAS_THRES_LOW_LSB	R/W	0x00	CAS Low Threshold	
0x203	CAS_MULT	R/W	0x11	CAS High Threshold Multiplier	
0x204	CAS_PRESENT	R/W	0x40	CAS Present Time	
0x205	CAS_ABSENT	R/W	0x20	CAS Absent Time	
0x206	CAS_STATUS	R	0x00	CAS Status	
0x207	CAS_MODE	R/W	0x10	CAS or Arbitrary Tone Detection Mode	
0x208	ATD_MAX_HFC	R/W	0x01	ATD Max High Frequency Cnt	
0x209	ATD_MAX_HFC_LSB	R/W	0x10	ATD Max High Frequency Cnt	
0x20A	ATD_MIN_HFC	R/W	0x01	ATD Min High Frequency Cnt	
0x20B	ATD_MIN_HFC_LSB	R/W	0x20	ATD Min High Frequency Cnt	
0x20C	ATD_MAX_LFC	R/W	0x01	ATD Max Low Frequency Cnt	
0x20D	ATD_MAX_LFC_LSB	R/W	0x61	ATD Max Low Frequency Cnt	
0x20E	ATD_MIN_LFC	R/W	0x01	ATD Min Low Frequency Cnt	
0x20F	ATD_MIN_LFC_LSB	R/W	0x6F	ATD Min Low Frequency Cnt	
0x210-0x241	ATD_HF_COEFF	R/W	0x00	ATD High Frequency IIR coefficient 0x210~0x241	
0x242-0x273	ATD_LF_COEFF	R/W	0x00	ATD Low Frequency IIR coefficient 0x242~0x273	
0x274	FSK_COEFF	R/W	0x00	FSK IIR Coefficient	
0x275	FSK_COEFF_ADDR	R/W	0x00	FSK IIR Coefficient Address	
0x276	CAS_MAX_HFC	R/W	0x01	CAS Max High Frequency Cnt	
0x277	CAS_MAX_HFC_LSB	R/W	0x0E	CAS Max High Frequency Cnt	

Addr.	Name	Mode	Value At Reset	Function	
0x278	CAS_MIN_HFC	R/W	0x01	CAS Min High Frequency Cnt	
0x279	CAS_MIN_HFC_LSB	R/W	0x21	CAS Min High Frequency Cnt	
0x27A	CAS_MAX_LFC	R/W	0x01	CAS Max Low Frequency Cnt	
0x27B	CAS_MAX_LFC_LSB	R/W	0x5D	CAS Max Low Frequency Cnt	
0x27C	CAS_MIN_LFC	R/W	0x01	CAS Min Low Frequency Cnt	
0x27D	CAS_MIN_LFC_LSB	R/W	0x75	CAS Min Low Frequency Cnt	
0x280	CPT_CTRL	R/W	0x07	CPT Control	
0x281	CPT_STATUS	R	0x00	CPT Status	
0x282	CPT_THRES_H_MSB	R/W	0x24	CPT High Threshold MSB	
0x283	CPT_THRES_H_LSB	R/W	0x00	CPT High Threshold LSB	
0x284	CPT_THRES_L_MSB	R/W	0x1A	CPT Low Threshold MSB	
0x285	CPT_THRES_L_LSB	R/W	0x00	CPT Low Threshold LSB	
0x286	CPT_ENERGY_MSB	R	0x00	CPT Energy MSB	
0x287	CPT_ENERGY_LSB	R	0x00	CPT Energy LSB	
0x290	TIME_CTRL	R/W	0x00	Timer Control	
0x292	TIME_TARG_MSB	R/W	0x00	Timer Target Counter MSB	
0x293	TIME_TARG_LSB	R/W	0x00	Timer Target Counter LSB	
0x294	TIME_CNT_MSB	R	0x00	Timer Current Counter MSB	
0x295	TIME_CNT_LSB	R	0x00	Timer Current Counter LSB	
0x300	AEC_CONFIG	R/W	0x96	AEC Configuration	
0x301	AEC_RESET	R/W	0xE8	AEC Reset	
0x302	AEC_EC_BELTA	R/W	0x03	AEC EC BELTA	
0x303	AEC_AS_COEFF	R/W	0x14	AEC AS Coefficient	
0x305	AEC_DT_LONG_TC	R/W	0x09	AEC Double Talk long term time constant	
0x306	AEC_DT_SHORT_TC	R/W	0xBB	AEC Double Talk short term time constant	
0x307	AEC_DT_HANGOVER_TIME_MSB	R/W	0x00	AEC Double Talk Detector Parameters	
0x308	AEC_DT_HANGOVER_TIME_LSB	R/W	0x20	AEC Double Talk Detector Parameters	
0x309	AEC_DT_DV_THRESH_MSB	R/W	0x19	AEC Double Talk Detector Parameters	
0x30A	AEC_DT_DV_THRESH_LSB	R/W	0x98	AEC Double Talk Detector Parameters	
0x30B	AEC_DT_LONG_THRESH_MSB	R/W	0x00	AEC Double Talk Detector Parameters	
0x30C	AEC_DT_LONG_THRESH_LSB	R/W	0x00	AEC Double Talk Detector Parameters	
0x30D	AEC_DT_SHORT_THRESH_MSB	R/W	0x10	AEC Double Talk Detector Parameters	

Addr.	Name	Mode	Value At Reset	Function	
0x30E	AEC_DT_SHORT_THRESH_LSB	R/W	0x10	AEC Double Talk Detector Parameters	
0x30F	AEC_DIVERGENCE	R/W	0x0F	AEC DIVERGENCE THRESHOLD	
0x310	AEC_VD_LONG_TC	R/W	0x09	AEC Voice Detect Long Time Constant	
0x311	AEC_VD_SHORT_TC	R/W	0xBB	AEC Voice Detect Short Time Constant	
0x312	AEC_VD_HANGOVER_TIME_MSB	R/W	0x00	AEC Voice Detect Parameters	
0x313	AEC_VD_HANGOVER_TIME_LSB	R/W	0x09	AEC Voice Detect Parameters	
0x314	AEC_VD_DEV_THRESHOLD_MSB	R/W	0x19	AEC Voice Detect Parameters	
0x315	AEC_VD_DEV_THRESHOLD_LSB	R/W	0x98	AEC Voice Detect Parameters	
0x316	AEC_VD_LONG_THRESH_MSB	R/W	0x19	AEC Voice Detect Parameters	
0x317	AEC_VD_LONG_THRESH_LSB	R/W	0x98	AEC Voice Detect Parameters	
0x318	AEC_VD_SHORT_THRESH_MSB	R/W	0x0B	AEC Voice Detect Parameters	
0x319	AEC_VD_SHORT_THRESH_LSB	R/W	0x38	AEC Voice Detect Parameters	
0x31A	AEC_CUT_OFF_POWER_MSB	R/W	0x19	AEC Voice Detect Parameters	
0x31B	AEC_CUT_OFF_POWER_LSB	R/W	0x98	AEC Voice Detect Parameters	
0x31D	AEC_VD_AVE_TC	R/W	0x0B	AEC VD Average Time Constant	
0x31E	AEC_VD_AVE_THRESH_MSB	R/W	0x00	AEC VD Average Threshold	
0x31F	AEC_VD_AVE_THRESH_LSB	R/W	0x00	AEC VD Average Threshold	
0x320	AEC_AS1_BUILD_UP_TIME	R/W	0x77	AEC AS1 Built Up Time	
0x321	AEC_AS1_MAX_ATTEN_MSB	R/W	0x1C	AEC AS1 Max Attenuation value(MSB)	
0x322	AEC_AS1_MAX_ATTEN_LSB	R/W	0xA8	AEC AS1 Max Attenuation value(LSB)	
0x323	AEC_AS2_BUILD_UP_TIME	R/W	0x77	AEC AS2 Built Up Time	
0x324	AEC_AS2_MAX_ATTEN_MSB	R/W	0x1C	AEC AS2 Max Attenuation value(MSB)	
0x325	AEC_AS2_MAX_ATTEN_LSB	R/W	0xAB	AEC AS2 Max Attenuation value(LSB)	
0x328	AEC_NS_POWER_ATTACK_TC	R/W	0xBB	AEC Noise Suppressor power attack time constant	
0x329	AEC_NS_ATTEN_TC	R/W	0xBB	AEC NS Attenuation Time Constant	
0x32A	AEC_NS_ACTIVE_THRESHOLD_MSB	R/W	0x03	AEC NS Active threshold (MSB)	
0x32B	AEC_NS_ACTIVE_THRESHOLD_LSB	R/W	0xE8	AEC NS Active threshold (LSB)	
0x32C	AEC_NO_VOICE_THRESH_MSB	R/W	0x14	AEC NO Voice Threshold MSB	
0x32D	AEC_NO_VOICE_THRESH_LSB	R/W	0x08	AEC NO Voice Threshold LSB	
0x330	AGC_THRES_MSB	R/W	0x20	AGC THRESHOLD MSB	
0x331	AGC_THRES_LSB	R/W	0x00	AGC THRESHOLD LSB	

Addr.	Name	Mode	Value At Reset	Function	
0x332	AGC_NOISE_THRES_MSB	R/W	0x03	AGC NOISE THRESHOLD MSB	
0x333	AGC_NOISE_THRES_LSB	R/W	0x20	AGC NOISE THRESHOLD LSB	
0x334	AGC_MAX_SG	R/W	0x02	AGC MAX SG	
0x335	AGC_LATK_TC	R/W	0xbb	AGC LATK TC Constant	
0x336	AGC_SATK_TC	R/W	0x09	AGC SATK Time Constant	
0x338	AEC_VD_SC_CTRL	R/W	0x00	AEC VD SC Control	
0x339	AEC_VD_SC_NORMAL_INDEX	R/W	0x00	AEC VD SC NORMAL INDEX	
0x33A	AEC_VD_SC_LOW_INDEX	R/W	0x00	AEC VD SC LOW INDEX	
0x33B	AEC_VD_SC_THRESH_MSB	R/W	0x10	AEC VD SC Threshold MSB	
0x33C	AEC_VD_SC_THRESH_LSB	R/W	0x00	AEC VD SC Threshold LSB	
0x33D	AEC_VD_SC_POWER_ATTACK_TC	R/W	0x07	AEC VD SC Power Attack Time Constant	
0x33E	AEC_VD_SC_GAIN_TC	R/W	0x07	AEC VD SC GAIN Time Constant	
0x340	AEC_DT_SHORT_TERM_POWER_MSB	R	NA	AEC DT Short Term Power MSB	
0x341	AEC_DT_SHORT_TERM_POWER_LSB	R	NA	AEC DT Short Term Power LSB	
0x342	AEC_DT_LONG_TERM_POWER_MSB	R	NA	AEC DT Long Term Power MSB	
0x343	AEC_DT_LONG_TERM_POWER_LSB	R	NA	AEC DT Long Term Power LSB	
0x344	AEC_DT_POWER_DEVIATION_MSB	R	NA	AEC DT POWER Deviation MSB	
0x345	AEC_DT_POWER_DEVIATION_LSB	R	NA	AEC DT POWER Deviation LSB	
0x346	AEC_DT_ACTIVE	R	NA	AEC DT Active	
0x348	AEC_VD_SHORT_TERM_POWER_MSB	R	NA	AEC VD Short Term Power MSB	
0x349	AEC_VD_SHORT_TERM_POWER_LSB	R	NA	AEC VD Short Term Power LSB	
0x34A	AEC_VD_LONG_TERM_POWER_MSB	R	NA	AEC VD Long Term Power MSB	
0x34B	AEC_VD_LONG_TERM_POWER_LSB	R	NA	AEC VD Long Term Power LSB	
0x34C	AEC_VD_POWER_DEVIATION_MSB	R	NA	AEC VD Power Deviation MSB	
0x34D	AEC_VD_POWER_DEVIATION_LSB	R	NA	AEC VD Power Deviation LSB	
0x34E	AEC_VD_ACTIVE	R	NA	AEC VD Active	
0x350	AEC_LRIN[15:8]	R	NA	AEC LRIN MSB	
0x351	AEC_LRIN[7:0]	R	NA	AEC LRIN LSB	
0x352	AEC_EOUT[15:8]	R	NA	AEC EOUT MSB	
0x353	AEC_EOUT[7:0]	R	NA	AEC EOUT LSB	
0x354	AEC_ASOUT15:8]	R	NA	AEC ASOUT MSB	
0x355	AEC_ASOUT[7:0]	R	NA	AEC ASOUT LSB	
0x356	AEC_ASIN[15:8]	R	NA	AEC ASIN MSB	
0x357	AEC_ASIN[7:0]	R	NA	AEC ASIN LSB	

Addr.	Name	Mode	Value At Reset	Function	
0x358	AEC_LSOUT[15:8]	R	NA	AEC LSOUT MSB	
0x359	AEC_LSOUT[7:0]	R	NA	AEC LSOUT LSB	
0x35C	AEC_MACRO_ADDR[15:8]	R/W	0x00	AEC Macro Address MSB	
0x35D	AEC_MACRO_ADDR[7:0]	R/W	0x00	AEC Macro Address LSB	
0x35E	AEC_MACRO_DATA[15:8]	R/W	NA	AEC Macro Data MSB	
0x35F	AEC_MACRO_DATA[7:0]	R/W	NA	AEC Macro Data LSB	
0x360	AGC_CTRL	R/W	0x00	AGC Control	
0x361	AGC_INIT_GAIN	R/W	0x88	AGC Initial Gain	
0x362	AGC_GAIN_HOLD	R/W	0x00	AGC Gain Hold	
0x363	AGC_INC_DEC	R/W	0x00	AGC INC and DEC	
0x364	AGC_ATK_DCY	R/W	0x00	AGC ATK DCY	
0x365	AGC_GAIN_READ	R	NA	AGC Gain Read	
0x366	AGC_STATE	R	NA	AGC State	
0x367	AGC_PWR_TC	R/W	0x99	AGC Power Time Constant	
0x368	AGC_PK_TC	R/W	0x00	AGC PK Time Constant	
0x369	AGC_PK_MSB	R	0x00	AGC PK MSB	
0x36A	AGC_PK_LSB	R	0x00	AGC PK LSB	
0x370	AGC_TARG_CLIP	R/W	0x73	AGC Target Clip MSB	
0x371	AGC_TARG_CLIP_LSB	R/W	0x33	AGC Target Clip LSB	
0x372	AGC_TARG_HI	R/W	0x68	AGC Target High MSB	
0x373	AGC_TARG_HI_LSB	R/W	0x00	AGC Target High LSB	
0x374	AGC_TARG_LO	R/W	0x66	AGC Target Low MSB	
0x375	AGC_TARG_LO_LSB	R/W	0x66	AGC Target Low LSB	
0x376	AGC_TARG_RB	R	0x00	AGC Target RB LSB	
0x377	AGC_TARG_RB_LSB	R	0x00	AGC Target RB LSB	
0x378	AGC_NOISE_BIAS	R/W	0x00	AGC Noise Bias	
0x379	AGC_NOISE_HI	R/W	0x00	AGC Noise High MSB	
0x37A	AGC_NOISE_HI_LSB	R/W	0x00	AGC Noise High LSB	
0x37B	AGC_NOISE_LO	R/W	0x00	AGC Noise Low MSB	
0x37C	AGC_NOISE_LO_LSB	R/W	0x00	AGC Noise Low LSB	
0x37D	AGC_NOISE_RB	R	0x00	AGC Noise [23:16]	
0x37E	AGC_NOISE_RB_2SB	R	0x00	AGC Noise [15:8]	
0x37F	AGC_NOISE_RB_LSB	R	0x00	AGC Noise [7:0]	

Addr.	Name	Mode	Value At Reset	Function	
0x380	LEC_CONFIG	R/W	0x96	LEC Configuration	
0x381	LEC_RESET	R/W	0xA8	LEC Reset	
0x382	LEC_EC_BELTA	R/W	0x03	LEC EC BELTA	
0x383	LEC_AS_COEFF	R/W	0x04	LEC AS Coefficient	
0x385	LEC_DT_LONG_TC	R/W	0x09	LEC Double Talk long term time constant	
0x386	LEC_DT_SHORT_TC	R/W	0xBB	LEC Double Talk short term time constant	
0x387	LEC_DT_HANGOVER_TIME_MSB	R/W	0x00	LEC Double Talk Detector Parameters	
0x388	LEC_DT_HANGOVER_TIME_LSB	R/W	0x20	LEC Double Talk Detector Parameters	
0x389	LEC_DT_DV_THRESH_MSB	R/W	0x19	LEC Double Talk Detector Parameters	
0x38A	LEC_DT_DV_THRESH_LSB	R/W	0x98	LEC Double Talk Detector Parameters	
0x38B	LEC_DT_LONG_THRESH_MSB	R/W	0x00	LEC Double Talk Detector Parameters	
0x38C	LEC_DT_LONG_THRESH_LSB	R/W	0x00	LEC Double Talk Detector Parameters	
0x38D	LEC_DT_SHORT_THRESH_MSB	R/W	0x10	LEC Double Talk Detector Parameters	
0x38E	LEC_DT_SHORT_THRESH_LSB	R/W	0x10	LEC Double Talk Detector Parameters	
0x38F	LEC_DIVERGENCE	R/W	0x0F	LEC DIVERGENCE THRESHOLD	
0x390	LEC_VD_LONG_TC	R/W	0x09	LEC Voice Detect Long Time Constant	
0x391	LEC_VD_SHORT_TC	R/W	0xBB	LEC Voice Detect Short Time Constant	
0x392	LEC_VD_HANGOVER_TIME_MSB	R/W	0x00	LEC Voice Detect Parameters	
0x393	LEC_VD_HANGOVER_TIME_LSB	R/W	0x09	LEC Voice Detect Parameters	
0x394	LEC_VD_DEV_THRESHOLD_MSB	R/W	0x19	LEC Voice Detect Parameters	
0x395	LEC_VD_DEV_THRESHOLD_LSB	R/W	0x98	LEC Voice Detect Parameters	
0x396	LEC_VD_LONG_THRESH_MSB	R/W	0x19	LEC Voice Detect Parameters	
0x397	LEC_VD_LONG_THRESH_LSB	R/W	0x98	LEC Voice Detect Parameters	
0x398	LEC_VD_SHORT_THRESH_MSB	R/W	0x10	LEC Voice Detect Parameters	
0x399	LEC_VD_SHORT_THRESH_LSB	R/W	0x38	LEC Voice Detect Parameters	
0x39A	LEC_CUT_OFF_POWER_MSB	R/W	0x19	LEC Voice Detect Parameters	
0x39B	LEC_CUT_OFF_POWER_LSB	R/W	0x98	LEC Voice Detect Parameters	
0x39D	LEC_VD_AVE_TC	R/W	0x0B	LEC VD Average Time Constant	
0x39E	LEC_VD_AVE_THRESH_MSB	R/W	0x00	LEC VD Average Threshold MSB	
0x39F	LEC_VD_AVE_THRESH_LSB	R/W	0x00	LEC VD Average Threshold LSB	
0x3A0	LEC_AS1_BUILD_UP_TIME	R/W	0x77	LEC AS1 Built Up Time	
0x3A1	LEC_AS1_MAX_ATTEN_MSB	R/W	0x1C	LEC AS1 Max Attenuation value(MSB)	
0x3A2	LEC_AS1_MAX_ATTEN_LSB	R/W	0xA8	LEC AS1 Max Attenuation value(LSB)	
0x3A3	LEC_AS2_BUILD_UP_TIME	R/W	0x77	LEC AS2 Built Up Time	

Addr.	Name	Mode	Value At Reset	Function	
0x3A4	LEC_AS2_MAX_ATTEN_MSB	R/W	0x1C	LEC AS2 Max Attenuation value(MSB)	
0x3A5	LEC_AS2_MAX_ATTEN_LSB	R/W	0xA8	LEC AS2 Max Attenuation value(LSB)	
0x3A8	LEC_NS_POWER_ATTACK_TC	R/W	0xBB	LEC Noise Suppressor power attack time constant	
0x3A9	LEC_NS_ATTEN_TC	R/W	0xBB	LEC NS Attenuation Time Constant	
0x3AA	LEC_NS_ACTIVE_THRESHOLD_MSB	R/W	0x03	LEC NS Active threshold (MSB)	
0x3AB	LEC_NS_ACTIVE_THRESHOLD_LSB	R/W	0xE8	LEC NS Active threshold (LSB)	
0x3B0	LEC_DT_SC_CTRL	R/W	0x00	LEC DT SC Control	
0x3B1	LEC_DT_SC_NORMAL_INDEX	R/W	0x00	LEC DT SC NORMAL INDEX	
0x3B2	LEC_DT_SC_LOW_INDEX	R/W	0x00	LEC DT SC Low INDEX	
0x3B3	LEC_DT_SC_THRESH_MSB	R/W	0x10	LEC DT SC Threshold MSB	
0x3B4	LEC_DT_SC_THRESH_LSB	R/W	0x00	LEC DT SC Threshold LSB	
0x3B5	LEC_DT_SC_POWER_ATTACK_TC	R/W	0x07	LEC DT SC Power Attack Time Constant	
0x3B6	LEC_DT_SC_GAIN_TC	R/W	0x07	LEC DT SC GAIN Time Constant	
0x3B8	LEC_VD_SC_CTRL	R/W	0x00	LEC VD SC Control	
0x3B9	LEC_VD_SC_NORMAL_INDEX	R/W	0x00	LEC VD SC NORMAL INDEX	
0x3BA	LEC_VD_SC_LOW_INDEX	R/W	0x00	LEC VD SC LOW INDEX	
0x3BB	LEC_VD_SC_THRESH_MSB	R/W	0x10	LEC VD SC Threshold MSB	
0x3BC	LEC_VD_SC_THRESH_LSB	R/W	0x00	LEC VD SC Threshold LSB	
0x3BD	LEC_VD_SC_POWER_ATTACK_TC	R/W	0x07	LEC VD SC Power Attack Time Constant	
0x3BE	LEC_VD_SC_GAIN_TC	R/W	0x07	LEC VD SC GAIN Time Constant	
0x3C0	LEC_DT_SHORT_TERM_POWER_MSB	R	NA	LEC DT Short Term Power MSB	
0x3C1	LEC_DT_SHORT_TERM_POWER_LSB	R	NA	LEC DT Short Term Power LSB	
0x3C2	LEC_DT_LONG_TERM_POWER_MSB	R	NA	LEC DT Long Term Power MSB	
0x3C3	LEC_DT_LONG_TERM_POWER_LSB	R	NA	LEC DT Long Term Power LSB	
0x3C4	LEC_DT_POWER_DEVIATION_MSB	R	NA	LEC DT POWER Deviation MSB	
0x3C5	LEC_DT_POWER_DEVIATION_LSB	R	NA	LEC DT POWER Deviation LSB	
0x3C6	LEC_DT_ACTIVE	R	NA	LEC DT Active	
0x3C8	LEC_VD_SHORT_TERM_POWER_MSB	R	NA	LEC VD Short Term Power MSB	
0x3C9	LEC_VD_SHORT_TERM_POWER_LSB	R	NA	LEC VD Short Term Power LSB	
0x3CA	LEC_VD_LONG_TERM_POWER_MSB	R	NA	LEC VD Long Term Power MSB	
0x3CB	LEC_VD_LONG_TERM_POWER_LSB	R	NA	LEC VD Long Term Power LSB	

Addr.	Name	Mode	Value At Reset	Function	
0x3CC	LEC_VD_POWER_DEVIATION_MSB	R	NA	LEC VD Power Deviation MSB	
0x3CD	LEC_VD_POWER_DEVIATION_LSB	R	NA	LEC VD Power Deviation LSB	
0x3CE	LEC_VD_ACTIVE	R	NA	LEC VD Active	
0x3D0	LEC_LRIN[15:8]	R	NA	LEC LRIN MSB	
0x3D1	LEC_LRIN[7:0]	R	NA	LEC LRIN LSB	
0x3D2	LEC_EOUT[15:8]	R	NA	LEC EOUT MSB	
0x3D3	LEC_EOUT[7:0]	R	NA	LEC EOUT LSB	
0x3D4	LEC_ASOUT15:8]	R	NA	LEC ASOUT MSB	
0x3D5	LEC_ASOUT[7:0]	R	NA	LEC ASOUT LSB	
0x3D6	LEC_ASIN[15:8]	R	NA	LEC ASIN MSB	
0x3D7	LEC_ASIN[7:0]	R	NA	LEC ASIN LSB	
0x3D8	LEC_LSOUT[15:8]	R	NA	LEC LSOUT MSB	
0x3D9	LEC_LSOUT[7:0]	R	NA	LEC LSOUT LSB	
0x3DC	LEC_MACRO_ADDR[15:8]	R/W	0x00	LEC Macro Address MSB	
0x3DD	LEC_MACRO_ADDR[7:0]	R/W	0x00	LEC Macro Address LSB	
0x3DE	LEC_MACRO_DATA[15:8]	R/W	NA	LEC Macro Data MSB	
0x3DF	LEC_MACRO_DATA[7:0]	R/W	NA	LEC Macro Data LSB	

新唐科技 NUVOTON INTELLECTUAL PROPERTY

7. DEVICE STATUS

7.1 Device Status Register

Whenever the ISD61S00 receives an SPI command it also returns its current status via MISO. The details of the status byte are shown below. For commands that are not reading digital data from the device this status byte is sent via MISO for every byte of data sent to the ISD61S00.

Table 7-1 Status Register Description

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

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

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

Whenever the ISD61S00 generates an interrupt the Interrupt Status registers hold flags that indicate what type of interrupt was generated. These flags will remain set until a READ_INT command clears them and the hardware interrupt pin (INTB). The interrupt enable registers IE0, IE1 (see section 8.5.3 8.5.4) mask whether internal functions generate interrupt events or not. Some interrupts require further servicing to remove the condition generating the interrupt, for instance a FIFO full or empty interrupt. If the condition is not serviced before a READ_INT, the device will immediately generate a new interrupt. The procedure to respond to an interrupt is the following:

1. READ_STATUS to determine what interrupt flags are set.
2. Service the interrupt appropriately,

3. READ_INT and check whether any new interrupts have occurred during the service routines. If new interrupt detected go to step 2. The READ_INT will clear the interrupt status.

Table 7-2 Interrupt Status Register Description

Interrupt Status Byte 1							
Bit 7	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
GPIO_INT	MPT_ERR	WR_FIN	CMD_ERR	OVF_ERR	CMD_FIN	ADDR_ERR	FULL_ERR

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

GPIO_INT	GPIO interrupt has occurred. A GPIO pin configured to produce an interrupt has toggled state.
MPT_ERR	Indicates a memory protection error. Digital access attempted for protected memory.
WR_FIN	Indicates a digital write command has finished writing to the flash memory.
CMD_ERR	An invalid command was sent to the device. Command was ignored because the command buffer was full, a voice macro was active or the device was not ready to respond to an erase command.
OVF_ERR	This error is generated if host illegally tries to read or write data while RDY/BSYB is low. It is also generated if a digital read or write attempts to read or write past the end of memory.
CMD_FIN	This bit indicates an interrupt was generated because a command finished executing. A CMD_FIN interrupt will be generated each time an audio play, record or voice macro finishes.
ADDR_ERR	Indicates an address error. This bit will be set to one if a PLAY_MSG@ command is sent at a non-valid header, a REC_MSG@ is sent at a non-blank memory location or an ERASE_MSG command is sent to a sector that is not the beginning of message.
FULL_ERR	Record Block is full. This bit will be set and an interrupt generated if a record command fills the memory.

Table 7-3 Interrupt Status Register Description

Interrupt Status Byte 2							
Bit 7	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
RNG_INT	CAS_INT	FSK_D	FSK_E	TONE_INT	DTMF_INT	CPT_INT	VD_INT

VD_INT	Voice Detection interrupt
CPT_INT	Call Progress Tone detector interrupt
DTMF_INT	The DTMF detector has generated an interrupt. This corresponds to a FIFO condition configured by the user.

TONE_INT	The tone or DTMF generator has generated an interrupt.
FSK_E	The FSK Encoder has generated an interrupt for either half empty or empty condition of FIFO.
FSK_D	The FSK Decoder has generated an interrupt for a FIFO condition or parity error.
CAS_INT	The CAS/Arbitrary Tone detector has detected a valid tone.
RNG_INT	The Ring Detect block has generated an interrupt.

Table 7-4 Timer interrupt and operation status

Timer interrupt and additional status							
Bit 7	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
TIMER_INT			DPLL_LOCK		PLL_LOCK	DECODE	COMP

TIMER_INT	Timer block has generated an interrupt.
DPLL_LOCK	Indicates lock of digital PLL for PCM slave mode synchronization.
PLL_LOCK	Status bit indicating that PLL is locked and valid clock is available.
DECODE	Status bit indicating that compressor is 1: decoding, 0: encoding
COMP	Status bit indicating that compressor is active.

8. FUNCTIONAL DESCRIPTION

ISD61S00 is a device targeted at applications requiring voice storage while interfacing to a PSTN telephony network. It provides high quality audio storage for record and playback and offers features of pre-recorded voice prompting and voice macros.

The acoustic side interface provides analog buffers to a microphone and audio output and incorporates on-chip acoustic echo cancellation to allow speakerphone functionality. Network side provides interface to the PSTN and includes a line echo cancellation network.

PCM data enters a mixer matrix whereby signals can be digitally gain controlled, mixed and passed between inputs and outputs. Additionally paths can be selected to monitor input for DTMF or FSK data.

All control and configuration is mapped into SPI accessible configuration registers. In addition these registers can be set from within Voice Macros allowing device to self configure from flash memory.

8.1 SPI Interface

Control of the ISD61S00 is via a four-wire SPI interface. This consists of an active low slave-select (SSB) signal to select the device, a serial clock (SCLK) a data input (Master Out Slave In - MOSI), and a data output (Master In Slave Out - MISO). In addition to the standard four wire interface a RDY/BSYB (ready, busy-bar) signal is supplied for data flow control needed in some transactions.

A SPI transaction begins on the falling edge of SSB and takes the form of the example below:

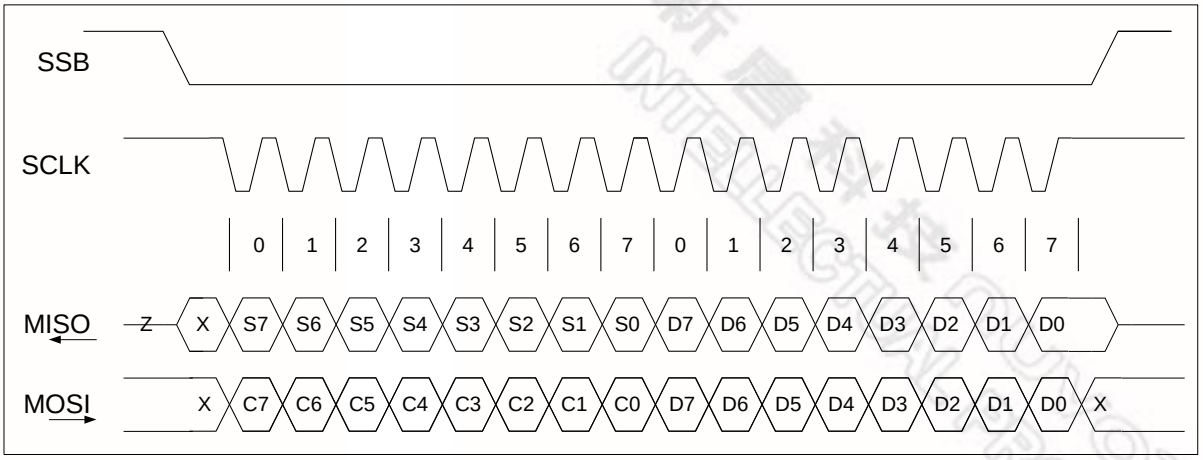


Figure 8-1 SPI Data Transaction

A command transaction begins with sending a command byte (C7-C0). The most significant bit (MSB – C7) is sent in first. During the shift in of the command byte, device status (S7-S0) is sent out MISO. After the command byte, depending upon the command sent, one or more bytes of data may follow. See Section 7.26 for details of SPI commands and data expected.

For a data writing commands RDY/BSYB is used to handshake data into the device. RDY/BSYB will change state after the rising edge of SCLK of the LSB of a byte transmission. When this occurs SCLK should be held high and data transmission paused until RDY/BSYB returns high indicating that device is ready to receive further data. The timing of this is shown in Figure 8-2 R/B Timing for SPI Writing Transactions

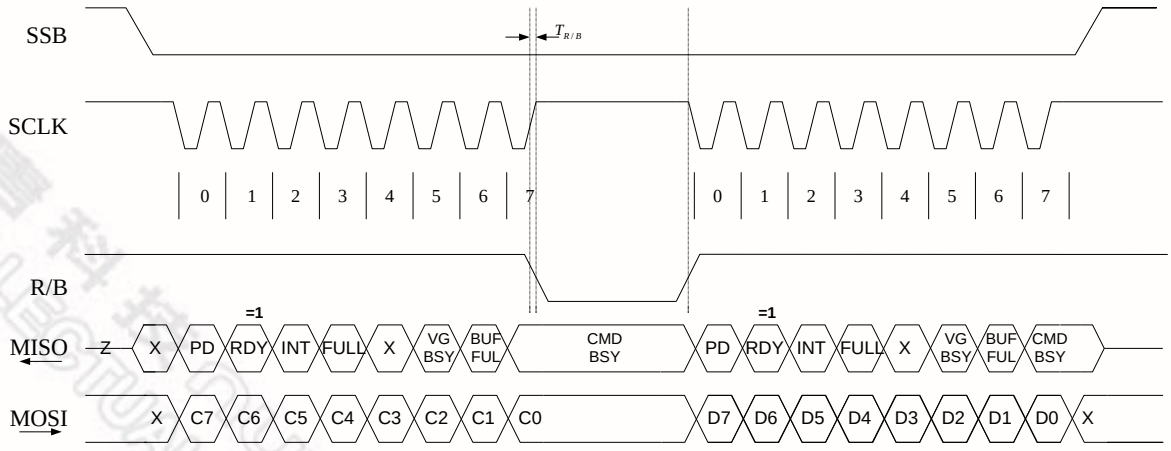


Figure 8-2 R/B Timing for SPI Writing Transactions

If host violates the request from RDY/BSYB, then the RDY bit of the status register for the next byte will reflect the state of RDY/BSYB. If host detects that the RDY bit of status is zero then the current byte of

data was not accepted by the ISD61S00. No further data will be accepted and the host must terminate the SPI transaction. For commands that read data from the part (for instance DIG_READ and SPI_READ) it is not possible to monitor the RDY bit of status so the RDY/BSYB signal must be monitored to know status. If RDY/BSYB is violated an interrupt is also generated to inform user that data was not accepted. The INT pin will go low shortly after the first violating edge of SCLK. In a data read operation, monitoring this interrupt is the only way of knowing that a RDY/BSYB violation has occurred apart from monitoring RDY/BSYB itself. Using this technique it is possible for a host that is IO limited to operate without monitoring RDY/BSYB.

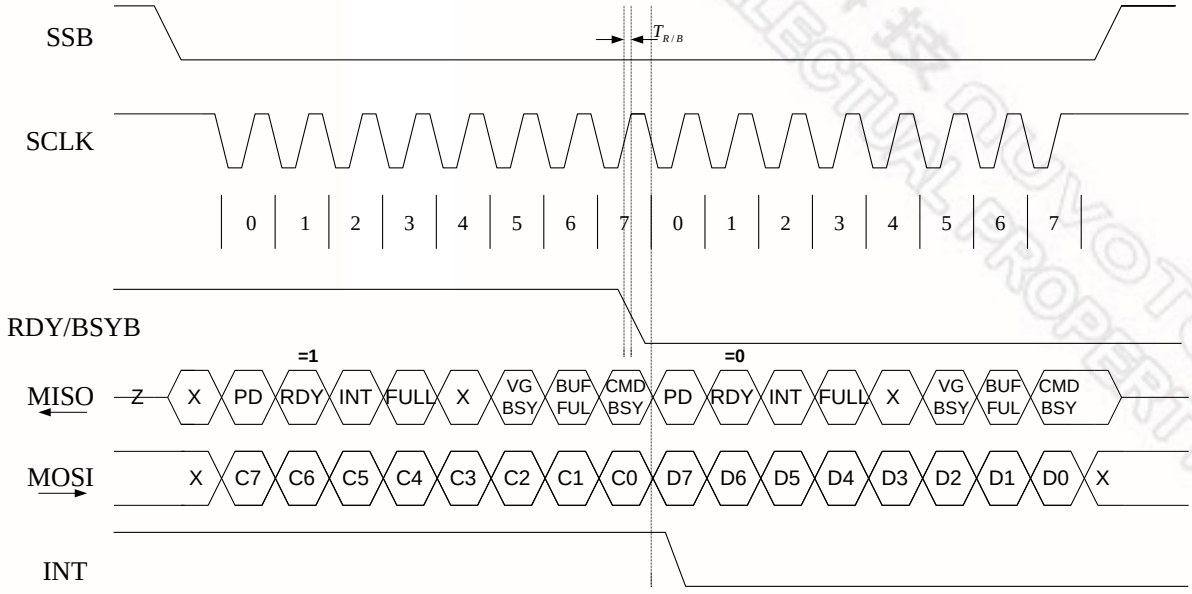


Figure 8-3 SPI Transaction Ignoring RDY/BSYB

8.2 Record and Playback Control

The ISD61S00 contains a compression and decompression engine that record and playback audio from internal flash memory. During recording, audio is re-sampled from the sample rate of the active audio path to the sample rate desired for storage. During playback audio data from the de-compression engine is re-sampled to that of the audio path.

8.2.1 COMP_CFG – Compression Configuration

The COMP_CFG register controls the sample rate and compression algorithm during message record operations. It can also override sample rate setting for playback by setting bit 0 of CFG1 high. SR[2:0]=COMP_CFG[7:5] controls the sample rate and CMP[4:0]=COMP_CFG[4:0] controls the compression. An explanation of these follows below.

Address	Access Mode	Value At Reset
0x000	R/W	0x64

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
SR[2:0]			CMP[4:0]				

CMP[4:0] Controls the compression algorithm used for recording. Details in Table 8-1.
SR[2:0] Controls the sample rate used for recording. Can also override the playback sample rate. Details in Table 8-2

Table 8-1 COMP_CFG Register Compression Type.

Compression Type	Bit rate (bits/sample)	CMP[4:0]	
		(Dec)	(Hex)
ADPCM	2	2	0x02
ADPCM	3	3	0x03
ADPCM	4	4	0x04
ADPCM	5	5	0x05
µ-Law	6	16	0x10
µ-Law	7	17	0x11
µ-Law	8	18	0x12
Dµ-Law	6	20	0x14
Dµ-Law	7	21	0x15
Dµ-Law	8	22	0x16
PCM	8	24	0x18
PCM	10	25	0x19
PCM	12	27	0x1B

Table 8-2 CFG0 Sample Rate Control

Sample Rate (kHz)	Code SR[2:0]	COMP_CFG[7:0]	
		(Dec)	(Hex)
4	0	0	0x00
5.333	1	32	0x20
6.4	2	64	0x40
8	3	96	0x60
10.6	4	128	0x80
12.8	5	160	0xA0
16	6	192	0xC0

The current operational mode of the ISD61S00 can also be queried by setting CFG1[4] and reading COMP_CFG. Under this condition, rather than reading back the configuration register, the result will be current audio path sample rate and compression scheme.

8.2.2 COMP_CTRL – Compression Control

Address	Access Mode	Value At Reset
0x001	R/W	0x00

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
FIFO Over-run	FIFO Under-run	LRMP	CFG0_READ	PAUSE	NRMP	SRSIL	SRCFG

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

SRCFG	Forces the sample rate to be set by the COMP_CFG register. By setting this bit to one sample rate contained in the recorded audio header is overridden.
SRSIL	Under normal circumstances, whenever a change in sample rate is detected between two consecutive messages a period of silence is automatically inserted. This is to prevent any transients in the signal path occurring as filter coefficients are changed. To turn off this silence insertion set this bit to one.
NRMP	Under normal conditions, if an audio playback finishes at a non-zero level the input to the signal path will be ramped to zero. This prevents a DC offset appearing on the output. To turn this feature off, for instance if a small audio sample is being looped, set this bit to one. Please note that **NRMP** and **LRMP** should not be set at the same time.
PAUSE	When this bit is set, Record or Playback operation will be paused. Clear this bit to resume the operation
CFG0_READ	When this bit is set, a read of COMP_CFG will result in the current operation mode of compression and sample rate rather than the setting of the COMP_CFG register. A write to COMP_CFG will still result in setting the register, but the contents of the register cannot be read back until CFG0_READ is set to zero.
LRMP	Set this bit to 1, the input to the signal path will not be ramped during the loop-play, but will be ramped to zero when the playback finishes or being stopped by a STOP or STOP_LP command. Please note that **NRMP** and **LRMP** should not be set at the same time.
FIFO Under-run	This is a read only register, that when high indicates that the audio FIFO has under-run, that is audio data was not present when required by the signal path. This is a normal condition at the end of a play command when compression is active. It can be used during a SPI_DAC operation to check whether data sent to ISD61S00 was corrupted. This signal is latched and is reset by writing a 1 followed by a 0 to CFG1[7], the FIFO Over-run bit.
FIFO Overrun	When read, a high indicates that the audio FIFO has over-run, that is audio data from the signal path could not be processed fast enough to keep audio integrity and data was lost. This is a normal condition at the end of a record command when compression has finished. It can be used during a SPI_ADC operation to check whether data received from the ISD161S00 was corrupted. This signal is latched and is reset by writing a 1 followed by a 0 to the register.

8.2.3 COMP_SRC – Compression Source

Address	Access Mode	Value At Reset
0x002	R/W	0x00

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Reserved	DECODE	SPI_IN	Reserved	Reserved	Reserved	ENCODE	SPI_OUT

SPI_OUT Output signal data to SPI. SPI_OUT and ENCODE cannot be both on at the same time. If DECODE is selected, the signal path is bypassed and ISD161S00 is ready for SPI playback operation. If DECODE is not set then the signal source is the audio path and a SPI ADC operation is selected. Use the SPI_RCV_AUD command to read the audio data out the SPI interface.

SPI_IN Input data from SPI. SPI_IN and DECODE cannot be both on at the same time. If ENCODE is set, then an SPI record operation is selected. If no audio data is sent to the signal path for a SPI DAC operation. Use the SPI_SND_AUD command to send audio data to the ISD61S00 from the SPI interface.

ENCODE Used in conjunction with a record operation. When selected, signal data is routed to the Audio compressor for compression. If SPI_IN is also selected, the signal path is bypassed and becomes ready for an SPI record operation using the SPI_SND_AUDIO command.

DECODE Used in conjunction with a play operation. When selected the signal path picks up data from the compressor and plays it to the corresponding outputs selected.

8.2.4 CLK_CTRL – Clock Control

Address	Access Mode	Value At Reset
0x003	R/W	0x03

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DPLL_RANGE			DPLL_EN	CODEC_FS	BCLK_RATE		

BCLK_RATE Determines the bit clock rate of the I2S interface in master mode.
0: 8.192MHz
1: 4.096MHz
2: 2.048MHz
3: 1.024MHz
4: 512kHz
5: 256kHz
6: 128kHz
7: 64kHz

CODEC_FS Determines the master sample rate of the audio CODEC. 0: 8kHz 1: 16kHz

DPLL_EN 1: Enable Digital PLL. Used when PCM interface in slave mode to synchronize master clock rate to an external frame sync sample rate on WS pin. To use this mode PLL must also be active.

DPLL_RANGE Lock-in range of digital PLL. Use smallest value that achieves locking. Range is approximately 0: 10ps, 1: 20ps, 3: 40ps, 7:120ps.

8.2.5 CFG17 – Update Mode

Address	Access Mode	Value At Reset
0x017	R/W	0x01

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PLL_LOCK	RB_SPI	Reserved	Reserved	DEC	COMPRESS	UPDATE	IMM

IMM	Immediate update mode. Registers 0x000-0x016 are double buffered. If IMM bit is set register values are active as soon as written.
UPDATE	If IMM bit is not set then registers 0x000-0x016 do not take effect until a 1 is written to this register
COMPRESS	Read only bit indicating that compression engine is active
DEC	Read only bit indicating that compression engine is 1: decoding 0: encoding
RB_SPI	This bit is set if users wish to monitor the RDY/BSYB handshake through SPI status rather than the hardware pin. When set it ensures that when conducting DIG_WRITE, SPI_SND_AUD and SPI_SND_DEC commands that the RDY bit of the status register is latched on SPI byte boundaries for correct read back.
PLL_LOCK	Read only status indicating that PLL is locked.

8.3 Clock Generation and PLL

The PLL may be used to multiply an external input clock reference frequency by a high resolution fractional number. To enable the use of the widest possible range of external reference clocks, the PLL block includes an optional divide-by-two prescaler for the input clock, a fixed divide-by-four scaler on the PLL output, and an additional programmable integer divider that is the Master Clock Prescaler.

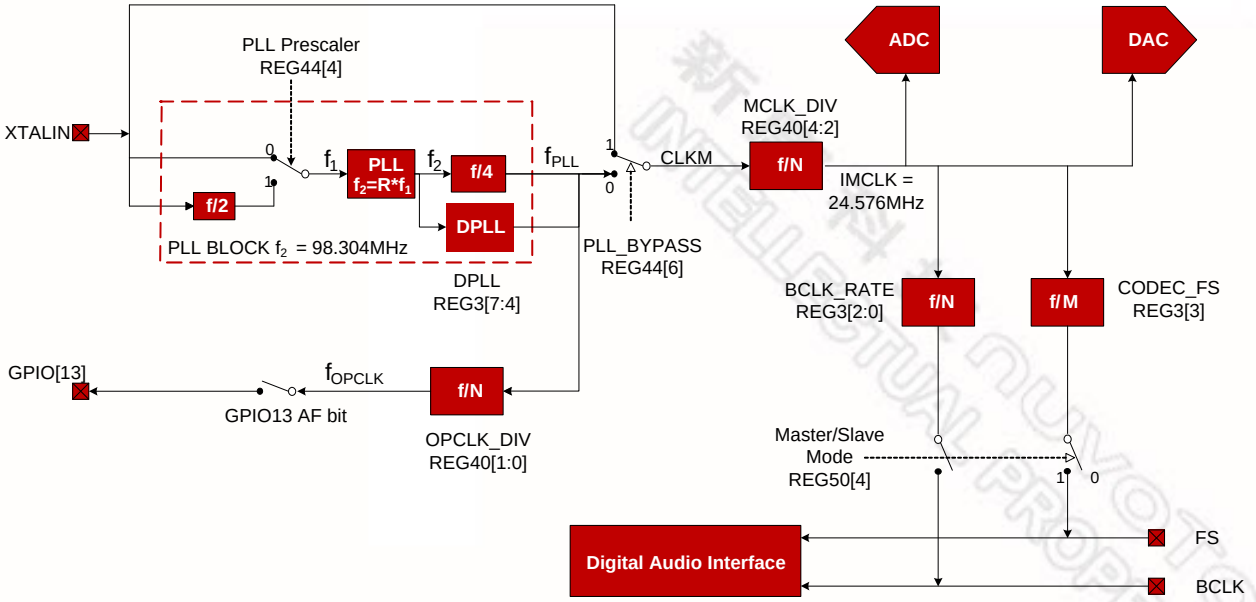


Figure 8-4 Clock Generation Block Diagram

The high resolution fraction for the PLL is the ratio of the desired PLL oscillator frequency and the reference frequency at the PLL input. This can be represented as R = f_2/f_1, with R in the form of a decimal number: xy.abcdefgh. To program the ISD61S00, this value is separated into an integer portion ("xy"), and a fractional portion. The fractional portion of the multiplier ("abcdefgh") is a value that when represented as a 24-bit binary number, most closely matches the exact desired multiplier factor.

To keep the PLL within its proper operating range, the integer portion of the decimal number ("xy"), must be any of the following decimal values: 6, 7, 8, 9, 10, 11, or 12. The input and output dividers outside of the PLL are used to scale frequencies as needed to keep the "xy" value within the required range.

Thus, for any given design, choose

R = f_2/f_1 = xy.abcdefgh decimal value

N = xy truncated to only the integer portion of the R value and N limited to decimal value 6, 7, 8, 9, 10, 11, or 12

K = (2^{24})(0.abcdefgh), rounded to the nearest whole integer value, then converted to hexadecimal 24-bit value

REG44[3:0] is set with the whole number integer portion (N) of the multiplier

REG41-REG43 is set collectively with the 24-bit fractional portion (K) of the multiplier

REG44[4] PLL Prescaler set as necessary

REG40[4:2] Master Clock Prescaler and BCLK Output Scaler set as necessary

8.3.1 Phase Locked Loop (PLL) Design Example

The target IMCLK rate is always 24.576MHz, which implies a PLL frequency f_2 of 98.304MHz.

In this example system design, there is already an available 12.000MHz clock from the USB subsystem. To reduce system cost, this clock will also be used for audio. Therefore, to use the 12MHz clock for

audio, the desired fractional multiplier ratio would be R = 98.304/12.000 = 8.192. This value meets the requirement that the "xy" whole number portion of the multiplier be in the range between 6 and 12.

To complete this portion of the design example, the integer portion of the multiplier is truncated to the value, 8. The fractional portion is multiplied by 2^{24}, as to create the needed 24-bit binary fractional value. The calculation for this is: $(2^{24})(0.192) = 3221225.472$. It is best to round this value to the nearest whole value of 3221225, or hexadecimal 0x3126E9. Thus, the values to be programmed to set the PLL multiplier whole number integer and fraction are:

```
R44   0xnm8    ; integer portion of fraction, (nm represents other settings in R44)
R41   0x31     ; highest order 8-bits of 24-bit fraction
R42   0x26     ; middle 8-bits of 24-bit fraction
R43   0xE9     ; lowest order 8-bits of 24-bit fraction
```

Below are additional examples of results for this calculation applied to commonly available clock frequencies.

MCLK (MHz)	PLL oscillator f_2 (MHz)	PLL Prescaler divider	Master Clock divider	Fractional Multiplier $R = f_2/f_1$	Integer Portion N (Hex)	Fractional Portion K (Hex)
12.0	98.3040	1	1	8.192000	8	3126E9
14.4	98.3040	1	1	6.826667	6	D3A06D
19.2	98.3040	2	1	10.240000	A	3D70A3
19.8	98.3040	2	1	9.929697	9	EE009E
24.0	98.3040	2	1	8.192000	8	3126E9
26.0	98.3040	2	1	7.561846	7	8FD526

Table 8-3: PLL Frequency Examples

8.3.2 PLL_CLK

Address	Access Mode	Value At Reset
0x040	R/W	0x20

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Reserved	MIC_CLK	OSR	MCLK_DIV			OPCLK_DIV	

OPCLK_DIV	Divisor of MCLK for optional GPIO clock output. $f_{OPCLK} = f_{PLL} / (OPCLK_DIV+1)$.
MCLK_DIV	Master clock scaler. Used to adjust clock to 24.576MHz for correct operation. Divisors are: 000=1, 001=1.5, 010=2, 011=3, 100=4, 101=6, 110=8, and 111=12.
OSR	Oversampling rate of DAC. 1: OSR = high (lower out of band noise), 0: OSR = low.
MIC_CLK	Microphone sampling clock frequency. 1: 64kHz, 0: 32kHz.

8.3.3 PLL_FRAC

Address	Access Mode	Value At Reset	Nominal Value
0x041-0x043	R/W	0x000000	

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
41	PLL_FRAC [23:16]							
42	PLL_FRAC [15:8]							
43	PLL_FRAC [7:0]							

PLL_FRAC The PLL fractional divider. This 24 bit number represents the fractional part of the PLL divider ratio R. Value of fractional part is PLL_FRAC ÷ 2^{24}

8.3.4 PLL_CTRL

Address	Access Mode	Value At Reset
0x044	R/W	0x88

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PLL_EN	PLL_BYPASS	Reserved	PLL_PRESCALER	PLL_DIV_N			

PLL_EN	Enable PLL
PLL_BYPASS	Bypass PLL and apply XTALIN clock directly to CLKM
PLL_PRESCALER	PLL Prescaler. 0: f_1 = XTALIN, 1: f_1 = XTALIN/2
PLL_DIV_N	Integer portion of PLL divisor ratio R. Must be in the range 6-12.

*To disable PLL_EN, first set (REG0x03) DPLL_EN = 0 and (REG0x44) PLL_BYPASS = 1. DPLL_EN cannot be enabled while PLL is disabled.

8.4 Digital Audio Interface Control

The ISD61S00 has a flexible digital audio interface to enable communication of digital serial audio. Communication is controlled by a bit clock and frame synchronization signal. The ISD61S00 has two physical transmit/receive channels, each capable of carrying two audio channels, to allow connection to multiple digital audio sources. Both these channels share common clock and synchronization. The clock and frame sync can be configured as outputs (ISD61S00 is master and generates clock and frame sync) or inputs (ISD61S00 is slave and accepts clock and frame sync). In slave mode the master clock input must either be synchronous to the frame sync signal or the internal digital PLL must be enabled to allow data synchronization.

The audio path of the ISD61S00 can generate/accept two channels of audio data. These can be routed to either PCM interface. In addition data can be looped from one PCM interface to the other.

The interface has a variety of modes. In I2S mode, PCM interface conforms to the I2S digital audio interface convention where left channel data is active when frame sync is low (WS=0) and right channel when frame sync is high (WS=1). In PCM mode data transfer is synchronized to the rising edge of the frame synch (WS) signal. Where data is placed for each channel is configurable through time slot control registers.

8.4.1 PCM_CFG

Address	Access Mode	Value At Reset
0x050	R/W	0x00

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
SCK_INV	TRI_EN	PCM_I2S	PCM_MODE	PCM1L	PCM1R	PCML	PCMR

PCMR	Enable PCM/I2S data out on right channel.
PCML	Enable PCM/I2S data out on left channel.
PCM1R	Enable PCM/I2S data out on right channel of PCM1 interface
PCM1L	Enable PCM/I2S data out on left channel of PCM1 interface
PCM_MODE	0: Master Mode, ISD61S00 generates SCK and WS. 1: Slave Mode, ISD61S00 accepts WS and SCK from host.
PCM_I2S	0: I2S mode. 1: PCM mode.
TRI_EN	Tristates the PCM bus after second half of LSB.
SCK_INV	0: Data clocks on rising edge of SCK. 1: data clocks on falling edge of SCK

The PCM/I2S bus uses GPIO pins 4-9 as listed below and is enabled by setting the PORTA_AF and PORTB_AF registers to enable the alternate functions of these pins. SCK frequency in master mode is controlled by CLK_CTRL (0x003) register (Section 8.2.4).

GPIO Port	GPIO Pin	I2S Name	Description	PCM Name	Master mode Direction	Slave Mode Direction
PORTA<4>	GPIO<4>	SDO	Serial Data Out	TX	O	O
PORTA<5>	GPIO<5>	WS	WS/Frame Sync	FS	O	I
PORTA<6>	GPIO<6>	SCK	Serial Bit Clock	BCLK	O	I
PORTA<7>	GPIO<7>	SDI	Serial Data In	RX	I	I
PORTB<0>	GPIO<8>	SDO1	Serial Data Out1	TX1	O	O
PORTB<1>	GPIO<9>	SDI1	Serial Data In1	RX1	I	I

In addition to I2S mode the interface can be configured into a generic PCM interface with control of where and how data is transmitted/received.

8.4.2 PCM_TSLOTL

Address	Access Mode	Value At Reset
0x051	R/W	0x01

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TSLOTL[7:0]							

TSLOTL[9:0] Left channel PCM data starts TSLOTL clocks after WS rising edge. Bits 9:8 are in PCM_COMP0 register.

8.4.3 PCM_TSLOTR

Address	Access Mode	Value At Reset
0x052	R/W	0x11

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TSLOTR[7:0]							

TSLOTR[9:0] Right channel PCM data starts TSLOTR clocks after WS rising edge. Bits 9:8 are in PCM_MODE register.

8.4.4 PCM_COMP0

Address	Access Mode	Value At Reset
0x053	R/W	0x00

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TSLOTR[9:8]		TSLOTL[9:8]		RX_COMP	RX_ULAW	TX_COMP	TX_ULAW

RX_COMP 1: Enable receive companding. 0: disable

RX_ULAW 1: Rx aLaw companding. 0: Rx uLaw companding.

TX_COMP 1: Enable transmit companding. 0: disable

TX_ULAW 1: Tx aLaw companding. 0: Tx uLaw companding.

8.4.5 PCM_TSLOTL1

Address	Access Mode	Value At Reset
0x054	R/W	0x01

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TSLOTL1[7:0]							

TSLOTL1[9:0] Left channel PCM1 data starts TSLOTL clocks after WS rising edge. Bits 9:8 are in PCM_COMP0 register.

8.4.6 PCM_TSLOTR1

Address	Access Mode	Value At Reset
0x055	R/W	0x11

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TSLOTR1[7:0]							

TSLOTR1[9:0] Right channel PCM1 data starts TSLOTR clocks after WS rising edge. Bits 9:8 are in PCM_MODE register.

8.4.7 PCM_COMP1

Address	Access Mode	Value At Reset
0x056	R/W	0x00

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TSLOTR1[9:8]		TSLOTL1[9:8]		RX_COMP1	RX_ULAW1	TX_COMP1	TX_ULAW1

RX_COMP1 1: Enable receive companding. 0: disable (PCM1)

RX_ULAW1 1: Rx aLaw companding. 0: Rx uLaw companding. (PCM1)

TX_COMP1 1: Enable transmit companding. 0: disable (PCM1)

TX_ULAW1 1: Tx aLaw companding. 0: Tx uLaw companding. (PCM1)

The following diagram illustrates the various signals that can be routed to the PCM/I2S interfaces.

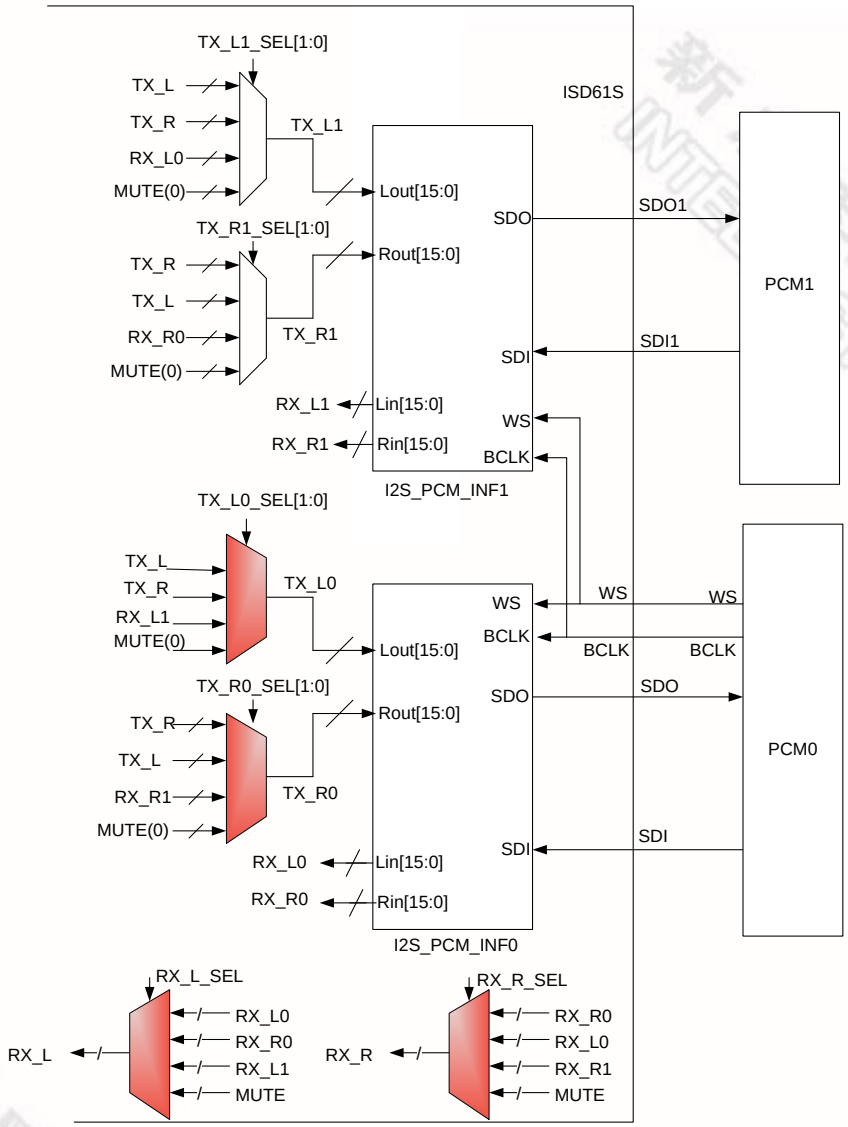


Figure 8-5 PMC/I2S signal routing

8.4.8 PCM_TX_SEL

Address	Access Mode	Value At Reset
0x057	R/W	0x00

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TX_R1_SEL		TX_L1_SEL		TX_R0_SEL		TX_L0_SEL	

TX_L0_SEL Select what audio to send on left channel of PCM bus

TX_R0_SEL Select what audio to send on right channel of PCM bus

TX_L1_SEL Select what audio to send on left channel of PCM1 bus

TX_R1_SEL Select what audio to send on right channel of PCM1 bus

8.4.9 PCM_RX_SEL

Address	Access Mode	Value At Reset
0x058	R/W	0x00

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Reserved		Reserved		RX_R_SEL		RX_L_SEL	

RX_L_SEL Select what audio to send to left channel signal path.

RX_R_SEL Select what audio to send to right channel of signal path

	TX_L0_SEL	TX_R0_SEL	TX_L1_SEL	TX_R1_SEL	RX_R_SEL	RX_L_SEL
0	TX_L	TX_R	TX_L	TX_R	RX_R0	RX_L0
1	TX_R	TX_L	TX_R	TX_L	RX_L0	RX_R0
2	RX_L1	RX_R1	RX_L0	RX_R0	RX_R1	RX_L1
3	MUTE	MUTE	MUTE	MUTE	MUTE	MUTE

8.5 GPIO and Interrupt Configuration

The ISD61S00 contains up to 17 GPIO pins. These pins can be configured as inputs or outputs, have configurable drive strength and pull-up pull-down capability. Some GPIO pins have an alternate optional function, such as I2S interface or ring detector input. The GPIO are divided into three ports: A, B and C. PORTA consists of GPIO<7:0>, PORTB consists of GPIO<15:8> and PORTC consists of GPIO<16>. The port functionality is controlled by the following configuration registers. Transitions on the GPIO pins can generate a GPIO_INT interrupt and also be configured to wake the chip up from a power-down state. The user must be aware that some GPIO pins have default alternate functions that must be enabled for correct operation (for internal memory devices these include PORTC[7:1] and for external memory devices PORTA[1:0] are used for memory access).

8.5.1 PORT_CFG

Address	Access Mode	Value At Reset	Nominal Value
0x00B	R/W	0x00	

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
RDET_IN		PORTC_H_INT	PORTC_L_INT	PORTB_H_INT	PORTB_L_INT	PORTA_H_INT	PORTA_L_INT

PORTA_L_INT	If this bit is set, and the corresponding PORTA_IE bit, then transitions on PORTA[3:0] will generate a GPIO wakeup interrupt.
PORTA_H_INT	If this bit is set, and the corresponding PORTA_IE bit, then transitions on PORTA[7:4] will generate a GPIO wakeup interrupt.
PORTB_L_INT	If this bit is set, and the corresponding PORTB_IE bit, then transitions on PORTB[3:0] will generate a GPIO wakeup interrupt.
PORTB_H_INT	If this bit is set, and the corresponding PORTB_IE bit, then transitions on PORTB[7:4] will generate a GPIO wakeup interrupt.
PORTC_L_INT	If this bit is set, and the corresponding PORTC_IE bit, then transitions on PORTC[3:0] will generate a GPIO wakeup interrupt.
PORTC_H_INT	Not used – set to zero.
RDET_IN[1:0]	These bits determine which GPIO port is used as the ring detector input. 0: GPIO<13>, 1: GPIO<14>, 2: GPIO<15>, 3: GPIO<16>

When the ISD61S00 goes into a power-down state the status of the GPIO ports are latched. If a GPIO pin, that has its interrupt enabled and wake-up enabled, toggles the chip will execute a wake-up event. The device will automatically execute Voice Macro 2 if present or, if this voice macro is blank, it will enter the power-up idle state and generate an interrupt to the host.

8.5.2 PORTx_IE

Address	Access Mode	Value At Reset	Nominal Value
0x00C-0x00E	R/W	0x00	

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PORTA_IE							
PORTB_IE							
PORTC_IE							

PORTA_IE	This register is the interrupt enable mask for PORTA. If the bit is set then that GPIO can generate a GPIO_INT event. Similarly for PORTB_IE and PORTC_IE

8.5.3 IE0

Address	Access Mode	Value At Reset	Nominal Value
0x014	R/W	0xFF	

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
GPIO_INT	MPT_ERR	WR_FIN	CMD_ERR	OVF_ERR	CMD_FIN	ADDR_ERR	FULL_ERR

The IE0 register enables the interrupt associated with the bit mask and corresponds to the interrupt status bits in STATUS_REG[1].

8.5.4 IE1

Address	Access Mode	Value At Reset	Nominal Value
0x015	R/W	0xFF	

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
RNG_INT	CAS_INT	FSK_D	FSK_E	TONE_INT	DTMF_INT	CPT_INT	VD_INT /TIMER_INT

The IE1 register enables the interrupt associated with the bit mask and corresponds to the interrupt status bits in STATUS_REG[2].

8.5.5 PORTA_DO

Address	Access Mode	Value At Reset	Nominal Value
0x019	R/W	0x00	

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PORTA_DO							

PORTA_DO — This register contains the data to be written out the GPIO pins of PORTA. Data will propagate to the pin if the corresponding PORTA_OE bit is enabled.

8.5.6 PORTA_OE

Address	Access Mode	Value At Reset	Nominal Value
0x01A	R/W	0x00	

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PORTA_OE							

PORTA_OE — This register contains output enable mask for the GPIO pins of PORTA. If a PORTA_OE bit is high data in the PORTA_DO register will propagate to the pin. If low the driver of the pin is in a tri-state condition.

8.5.7 PORTA_PE

Address	Access Mode	Value At Reset	Nominal Value
0x01B	R/W	0xFF	

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PORTA_PE							

PORTA_PE — This register contains the pull enable mask for the GPIO pins of PORTA. If a PORTA_PE bit is high then a pull-up or pull-down is connected to the pin depending

on the state of PORTA_PS.

8.5.8 PORTA_DI

Address	Access Mode	Value At Reset	Nominal Value
0x01C	R	0xFF	

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PORTA_DI							

PORTA_DI This read only register reflects the state of the GPIO pins of PORTA.

8.5.9 PORTA_PS

Address	Access Mode	Value At Reset	Nominal Value
0x01D	R/W	0xFF	

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PORTA_PS							

PORTA_PS This register contains the pull select mask for the GPIO pins of PORTA. If a PORTA_PS bit is high then a pull-up is selected or if low a pull-down.

8.5.10 PORTA_DS

Address	Access Mode	Value At Reset	Nominal Value
0x01E	R/W	0x00	

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PORTA_DS							

PORTA_DS This register selects the drive strength of PORTA pins. A high selects high drive a low configures a normal drive capability. Refer to Section 3 for drive value.

8.5.11 PORTA_AF

Address	Access Mode	Value At Reset	Nominal Value
0x01F	R/W	0x03	

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PORTA_AF							

PORTA_AF This register selects the alternate function of the GPIO pin.

The ports B and C have similar control registers with the following addresses:

Register	Address (HEX)
PORTA_DO	019
PORTA_OE	01A
PORTA_PE	01B
PORTA_DI	01C
PORTA_PS	01D
PORTA_DS	01E
PORTA_AF	01F
PORTB_DO	030
PORTB_OE	031
PORTB_PE	032
PORTB_DI	033

Register	Address (HEX)
PORTB_PS	034
PORTB_DS	035
PORTB_AF	036
PORTC_DO	037
PORTC_OE	038
PORTC_PE	039
PORTC_DI	03A
PORTC_PS	03B
PORTC_DS	03C
PORTC_AF	03D

新唐科技 NUVOTON INTELLECTUAL PROPERTY

The following ports have alternate functions:

Port	GPIO Pin	Alternate Function	Description
PORTA<0>	GPIO<0>	DO	DO pin for external memory access
PORTA<1>	GPIO<1>	CLK	CLK pin for external memory access
PORTA<4>	GPIO<4>	SDO	I2S Serial Data Out
PORTA<5>	GPIO<5>	WS	I2S WS
PORTA<6>	GPIO<6>	SCK	I2S Serial Clock
PORTA<7>	GPIO<7>	SDI	I2S Serial Data In
PORTB<0>	GPIO<8>	SDO1	PCM1 Tx Data
PORTB<1>	GPIO<9>	SDI1	PCM1 Rx Data

8.6 16 bit Indirect Index register R0-R7

- 16 bit R0 ~ R7 register store the VP/VM indexes for the indirect playback commands. Indirect Playback commands include:
  - PLAY_VP@Rn
  - PLAY_VP_LP@Rn
  - EXE_VM@Rn

For example: If R0=0x0020, then a PLAY_VP@R0 command will actually play VP @ index 0x0020; a - EXE_VM@R0 command will actually execute VM @ index 0x0020

R0-R7 Register Mapping							
Bit 7	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
CFG20 R0[7:0]							
CFG21 R0[15:8]							
CFG22 R1[7:0]							
CFG23 R1[15:8]							
CFG24 R2[7:0]							
CFG25 R2[15:8]							
CFG26 R3[7:0]							
CFG27 R3[15:8]							
CFG28 R4[7:0]							
CFG29 R4[15:8]							
CFG2A R5[7:0]							
CFG2B R5[15:8]							

CFG2C R6[7:0]
CFG2D R6[15:8]
CFG2E R7[7:0]
CFG2F R7[15:8]

8.7 Microphone interface and Auxiliary Interface

The ISD61S00 contains a fully integrated programmable microphone interface. No external components other than the microphone are required to operate the circuit. The microphone interface can operate in three modes: 1) Microphone Voltage gain mode; 2) Microphone Current gain mode; 3) Auxiliary input mode. For the Current gain mode an internal or external resistor can be selected to determine the gain. For the Auxiliary input mode it is recommended external resistors are used, though optionally the internal gain resistor can be selected.

The interface modes above can be selected with register MIC_MODE[2:0] of register MIC_CTRL. The microphone block is powered down until MIC_EN bit of MIC_CTRL is set to one.

In order to minimize the audible pops and clicks, it is not advised to change the MIC_GAIN[7:4] in real time but preferably mute the digital signal path prior to any gain changes.

8.7.1 MIC_CTRL

Address	Access Mode	Value At Reset
0x006	R/W	0x00

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
MIC_GAIN				MIC_EN	MIC_MODE		

MIC_MODE[2:0] Sets AC input amplifier mode of operation (voltage mode, current mode, external resistor, internal resistor)

Table 8-4 Microphone and Auxiliary mode settings

MIC_MODE[2:0]	Operation mode
000	Microphone Voltage-gain mode with internal gain resistor
001	Not allowed
010	Microphone Current-gain mode with internal gain resistor
011	Microphone Current-gain mode with external gain resistor
100	Auxiliary input mode with internal gain resistor
101	Auxiliary input mode with external gain resistor

MIC_GAIN[7:4] Set microphone gain in voltage-gain mode. Adjustable voltage gain range: 14dB~37.6dB, 16 steps linear in dB with step size of approximately 1.5dB.

Table 8-5 Microphone Gain settings

MIC_GAIN[7:4]	Gain(dB)	Boosted	MIC_GAIN[7:4]	Gain(dB)	Boosted
0000	37.65	43.1	1000	24.80	30.5
0001	35.98	41.5	1001	23.17	28.7
0010	34.33	40	1010	21.53	27.05
0011	32.70	38.2	1011	19.95	25.5
0100	31.06	36.6	1100	18.54	24.2
0101	29.47	35.05	1101	16.86	22.4
0110	27.89	33.5	1110	15.52	21.2
0111	26.38	31.88	1111	13.95	19.5

8.7.2 MIC_BIAS

Address	Access Mode	Value At Reset
0x007	R/W	0x00

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
MIC_RES				BOOST	MIC_BIAS		

BOOST Boost microphone voltage gain.

MIC_BIAS[2:0] Set microphone bias voltage reference

Table 8-6 Microphone Bias settings

MIC_BIAS[2:0]	Vref (V)
000	1.992
001	2.191
010	2.434
011	2.738
100	1.218
101	1.253
110	1.512
111	1.754

MIC_RES[7:4] Set microphone bias resistor

Table 8-7 Microphone Resistor settings

MIC_RES[7:4]	RL (kΩ)	Code	RL (kΩ)
0100	Open	1100	1.25
0101	10	1101	1.11
0110	5	1110	1
0111	3.33	1111	0.91
0000	2.5	1000	0.83
0001	2	1001	0.77
0010	1.67	1010	0.71
0011	1.42	1011	0.67

The basic operation of the voltage gain mode is shown in Figure 8-6 Microphone Voltage Gain Mode . The microphone is connected to the pins MCP and MCGND. The pin MCO is connected to the output of the microphone amplifier and can be used for monitoring the AC level. The voltage gain is set by register MIC_GAIN[7:4]. This register provides a gain range from 14dB to 38dB. In addition the BOOST bit of MIC_BIAS[3] can be enabled to add an additional 6dB voltage gain. The gain is set by a ratio of internal resistors, providing accurate gain control. The pin MCP also supplies the bias reference for the microphone. The bias consists of a programmable resistor and a programmable low noise voltage reference. The programmable resistor is set by register MIC_RES[7:4] and can be set to open and 670 Ohm to 10kOhm. The programmable voltage reference is set by register MIC_BIAS[2:0] and can be set from 1.22Volt to 2.74Volt.

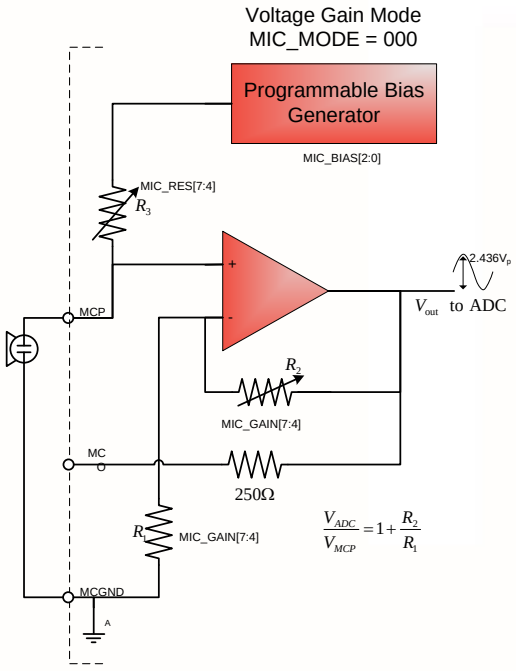


Figure 8-6 Microphone Voltage Gain Mode

For higher gain configurations, the **current gain mode** can be used. Figure 8-7 Microphone Current Gain Mode

shows how the current gain mode is used. The current gain mode uses the same programmable microphone bias voltage and resistor as the voltage gain mode. The gain is set by either the internal gain resistor or an external resistor, depending on the MIC_MODE setting. Since the current gain mode is using a single resistor, the gain accuracy is limited. However, large gain can be achieved. Note that a 250Ohm ESD protection resistor is connected to the MCO pin. This resistor should be considered when calculating the gain.

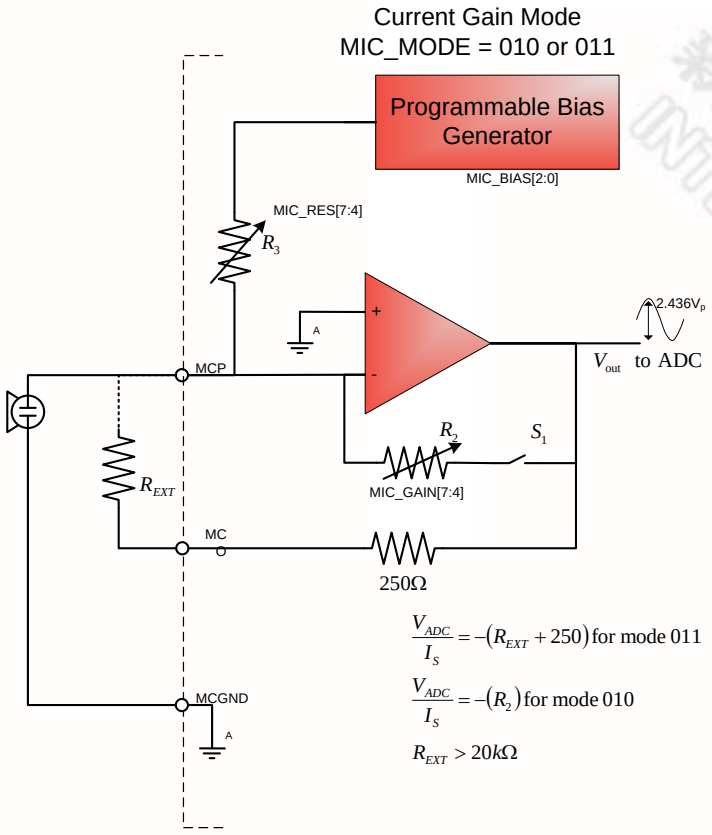


Figure 8-7 Microphone Current Gain Mode

The value of R_2 feedback resistor for mode 010 is shown below. It fixes the transimpedance gain. The resistor accuracy is +/-15%.

MIC_GAIN[7 :4]	R_2 (kΩ)
0	300
1	247
2	203
3	167
4	137
5	113
6	93
7	76
8	63
9	52
10	42
11	35
12	29

13	24
14	19
15	16

The voltage gain from MCP to MCO can be expected in the range of 29 to 55dB with a standard electret microphone and a microphone bias resistor set to 10K Ohm (MIC_RES[7:4]=0x05)

The voltage gain from MCP to MCO can be expected in the range of 26 to 49dB with a standard electret microphone and a microphone bias resistor set to 0.67K Ohm (MIC_RES[7:4]=0x0B)

One should note that the gain depends on the microphone internal characteristics and part number.

The following figure is an example and should not be taken as an exact list of gains.

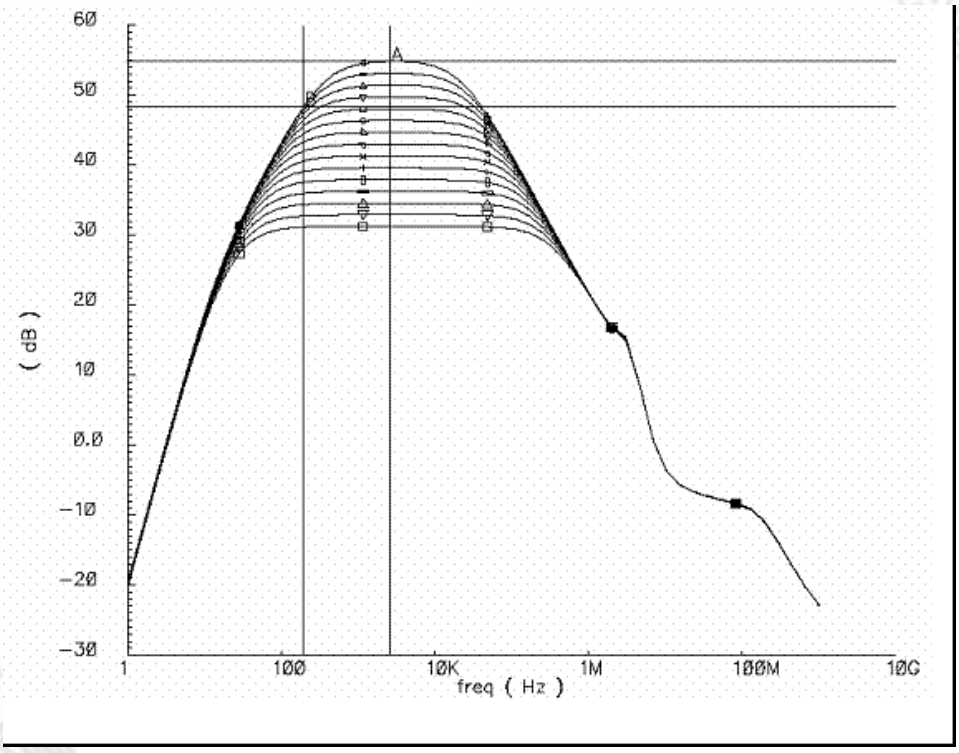


Figure 8-8 Current Mode Gain with Electret MIC model

For non-microphone applications one or more Auxiliary inputs can be connected to the MCP pin as shown in Figure 8-9 Auxiliary Input Mode

below. The MIC_RES register should be set to open in order to disconnect the microphone bias. For the gain setting it is advised to use external gain resistors only for optimal matching and accuracy. The 250Ohm ESD protection resistor should be considered again when calculating the gain. Note that for this mode the MCGND pin is tied to the external supply ground. A clean ground reference should be used for this.

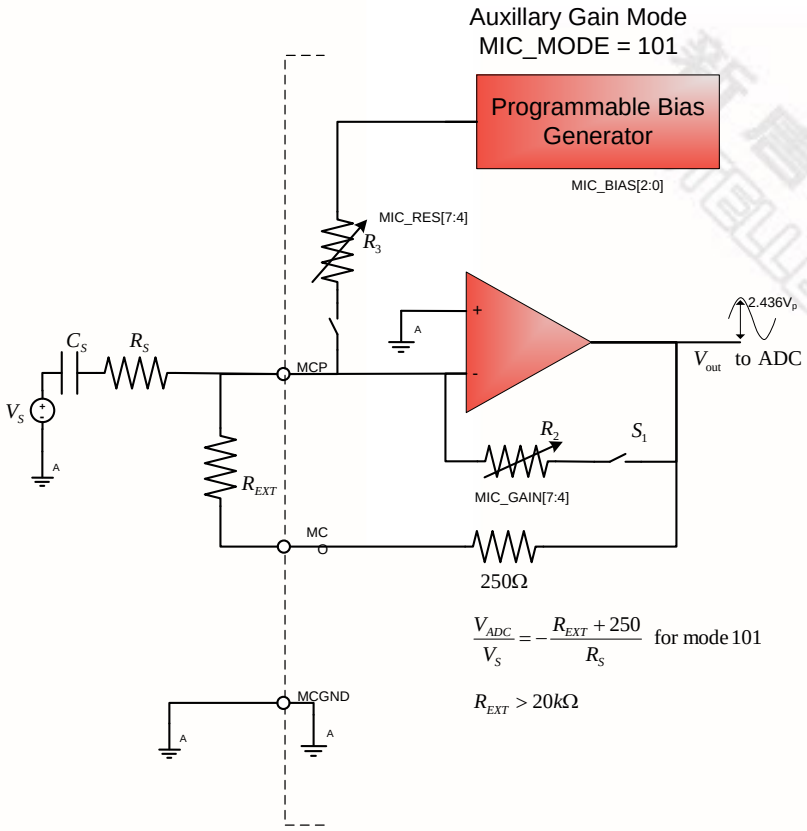


Figure 8-9 Auxiliary Input Mode

8.8 PSTN Analog Input

The PSTN analog input consists of two multiplexed pins to allow on-hook CID snooping along with normal off-hook operation. Input amplifier consists of two stages with independently settable gains as described in Table 8-8.

The input source of each stage is controlled by a multiplexor set by registers described in Table 8-9. Stage one input can be selected from the TI1 input, the TI2 input or a loop-back of the PO output. Stage two inputs can be stage one allowing cascading of gains, or stage one can be bypassed and inputs of TI1, TI2 or TI3 can be selected. Note that TI3 is not necessary available as described in the Pin Configuration (section 3). The two gain stages together can provide gains from 0dB to 42dB. In addition, power control for this amplifier is controlled by the TI_EN bit of the TI_CTRL register.

Inputs should be capacitively coupled into the TIxN and TIxP pins. The gain stage feeds the Line ADC which has a full scale range of Avdd/(2*1.41)V_{rms} or Avdd/2 V_p.

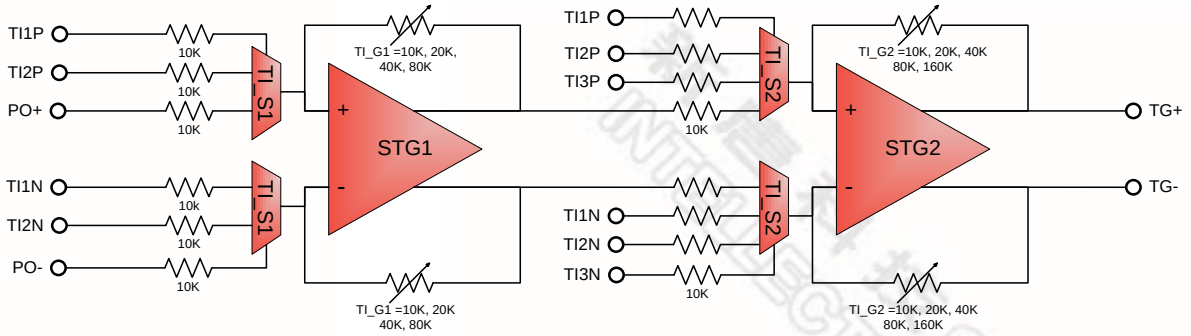


Figure 8-10 TI Input Amplifier

8.8.1 TI_GAIN

Address	Access Mode	Value At Reset
0x004	R/W	0x00

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
RESERVED			TI_G2			TI_G1	

TI_G1[1:0] Set stage 1 TI amplifier gain.

TI_G2[2:0] Set stage 2 TI amplifier gain.

8.8.2 TI_CTRL

Address	Access Mode	Value At Reset
0x005	R/W	0x00

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TI_EN	RESERVED			TI_S2		TI_S1	

TI_S1[1:0] Select input of stage 1 TI amplifier.

TI_S2[1:0] Select input of stage 2 TI amplifier.

TI_EN Enable (power up) TI amplifier.

First stage selection

TI_S1[1]	TI_S1[0]	input selected	
0	0	none	Don't use
0	1	TI1	Select TI1 pins
1	0	TI2	Select TI2 pins
1	1	PO*	Select PO pins

Second stage selection

TI_S2[1]	TI_S2[0]	input selected	
0	0	Output of stage 1	default
0	1	TI1	Select TI1 pins
1	0	TI2	Select TI2 pins
1	1	TI3	Select TI3 pins

* When this mode is selected the PO+ PO- pins are internally connected to the input of the PSTN analog interface.

First stage gain settings

TI_G1[1]	TI_G1[0]	gain	unit
0	0	0	dB
0	1	6	dB
1	0	12	dB
1	1	18	dB

Second stage gain settings

TI_G2[2]	TI_G2[1]	TI_G2[0]	gain	unit
0	0	0	0	dB
0	0	1	6	dB
0	1	0	12	dB
0	1	1	18	dB
1	0	0	24	dB

Table 8-8 TI Gain Settings

TI_G1[1:0]		Gain [dB]	TI_G2[2:0]		Gain [dB]
Bin	Hex		Bin	Hex	
00	0	0 dB	000	0	0 dB
01	1	6 dB	001	1	6 dB
10	2	12 dB	010	2	12 dB
11	3	18 dB	011	3	18 dB
			100	4	24 dB

Table 8-9 TI MUX Settings

TI_S1[1:0]		Input	TI_S2[1:0]		Input
Bin	Hex		Bin	Hex	
00	0	None	00	0	STG1
01	1	TI1	01	1	TI1
10	2	TI2	10	2	TI2
11	3	PO	11	3	TI3

8.9 Analog Outputs

Analog outputs consist of a differential speaker driver capable of driving 8Ω and a differential line driver for the PSTN interface capable of driving 120Ω. These buffers are connected to the two DAC paths incorporating digital gain control. The ANA_OUT register controls the analog function of these two drivers allowing power control via SPK_EN and PO_EN and gain setting (including MUTE) via PO_GAIN and SPK_GAIN. In addition a single-ended AUX output capable of driving a load of 1uF plus 47KΩ in series is available. The AUX driver full scale output voltage is half the one of the PO and SPK drivers.

When the PO and SPK drivers are power down their outputs are tri-stated. When the AUX driver is power down tri-state is achieved by setting the register TRI_STATE to 1.

When enabled, the DC level on the analog outputs is typically AVDD/2.

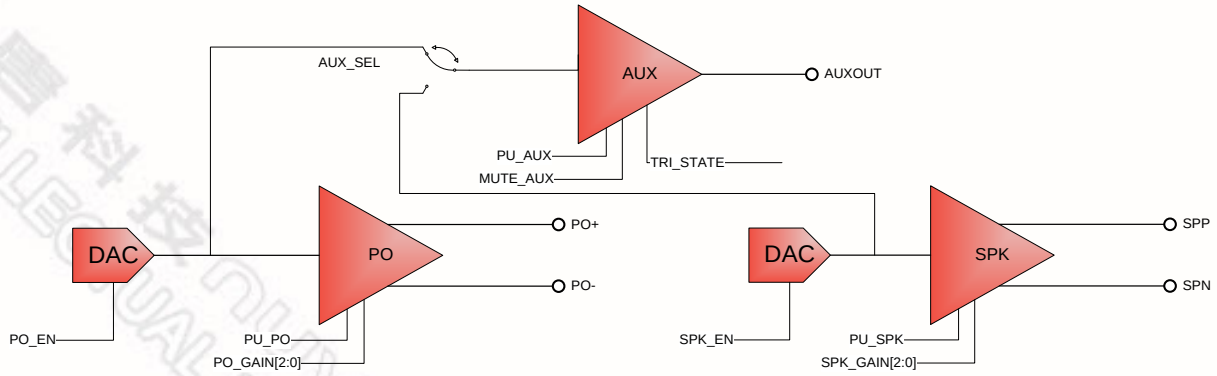


Figure 8-11 Analog Output

8.9.1 ANA_OUT

Address	Access Mode	Value At Reset
0x008	R/W	0x00

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PO_EN	PO_GAIN			SPK_EN	SPK_GAIN		

SPK_GAIN Set speaker driver gain

SPK_EN = 0 Disable (power down) speaker DAC
= 1 Enable speaker DAC

PO_GAIN Set line driver gain

PO_EN = 0 Disable (power down) line DAC
= 1 Enable line DAC

SPK_GAIN[2:0]	Gain(dB)
0XX	MUTE
100	0
101	0
110	3
111	-3

PO_GAIN[5:4]	Gain(dB)
0XX	MUTE
100	0
101	0
110	3
111	-3

8.9.2 ANA_CTRL

Address	Access Mode	Value At Reset
0x009	R/W	0x00

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PU_PO	PU_SPK	AUX_SEL	TRI_STATE	Reserved		PU_AUX	MUTE_AUX

MUTE_AUX = 0 Mute auxiliary buffer
= 1 Un-mute auxiliary buffer.

PU_AUX = 0 power down
= 1 power up

TRI_STATE = 0 Default
= 1 Disconnect any impedances from the AUXOUT output

AUX_SEL = 0 PSTN DAC is selected (default)
= 1 SPEAKER DAC is selected

新唐科技 NUVOTON INTELLECTUAL PROPERTY

PU_SPK	= 0	power down
	= 1	power up
PU_PO	= 0	power down
	= 1	power up

8.10 DTMF Detection

Dual Tone Multi Frequency (DTMF) signals consist of a sum of two tones: a low tone of 697Hz, 770Hz, 852Hz or 941Hz (row tone) and a high tone of 1209Hz, 1336Hz, 1477Hz or 1633Hz (column tone) as shown in Table 8-10. The DTMF detector filters the incoming signal and separates it into row and column tones, as shown in Figure 8-12. When valid data is detected, the resulting decoded data is pushed onto a FIFO which can be read by the host through the SPI interface.

When DTMF detection is enabled channel data is scanned for DTMF tones. Three programmable timer registers – Present Detect Time (DTMF_PDT), Acceptance Time (DTMF_ACCT), and Absent Detect Time (DTMF_ADT) – and the frequency deviation (DTMF_FREQ_DEV) and threshold (DTMF_THRES) registers allow the user to configure the sensitivity of the DTMF detector, as shown in Figure 8-13 and described below.

The DTMF detector performs a frequency estimation calculation to determine the frequencies present in the channel data input. Once the input signal has been determined to carry energy exceeding DTMF_THRES at one of the row (column) frequencies for DTMF_PDT within the error tolerance specified by DTMF_FREQ_DEV, the row (column) detect flag is asserted.

When both the row and column detect flags have been asserted, the DTMF_STATE goes to 1. Once the DTMF_STATE has been high for DTMF_ACCT, the signal is determined to be a valid DTMF signal. At this point, DTMF_RDY is asserted and the received data is decoded from the row and column frequencies and pushed onto the FIFO. The DTMF_STATE and DTMF_RDY status flags may be read from the DTMF_RX_DATA status register.

When the tone is outside of the tolerance specified by DTMF_FREQ_DEV for DTMF_ADT, the row (column) detect flag is de-asserted.

When a DTMF tone is detected the ISD61S00 can be configured to generate an interrupt to the host processor for service.

DTMF detection is available with the 8KHz sampling rate.

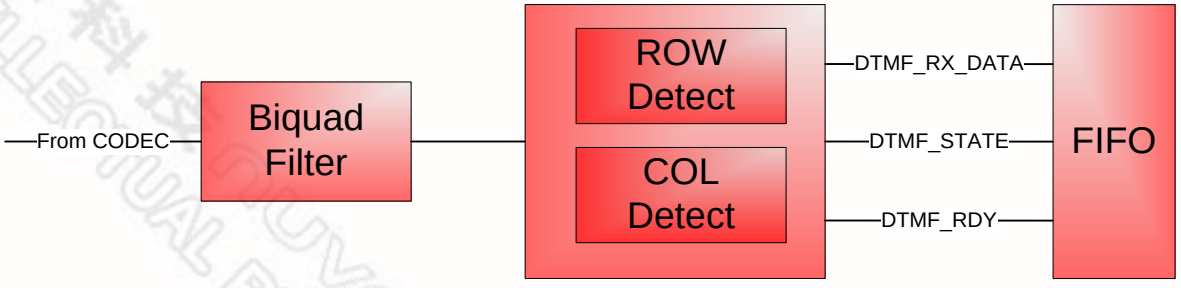


Figure 8-12 DTMF Detector - Functional Block Diagram

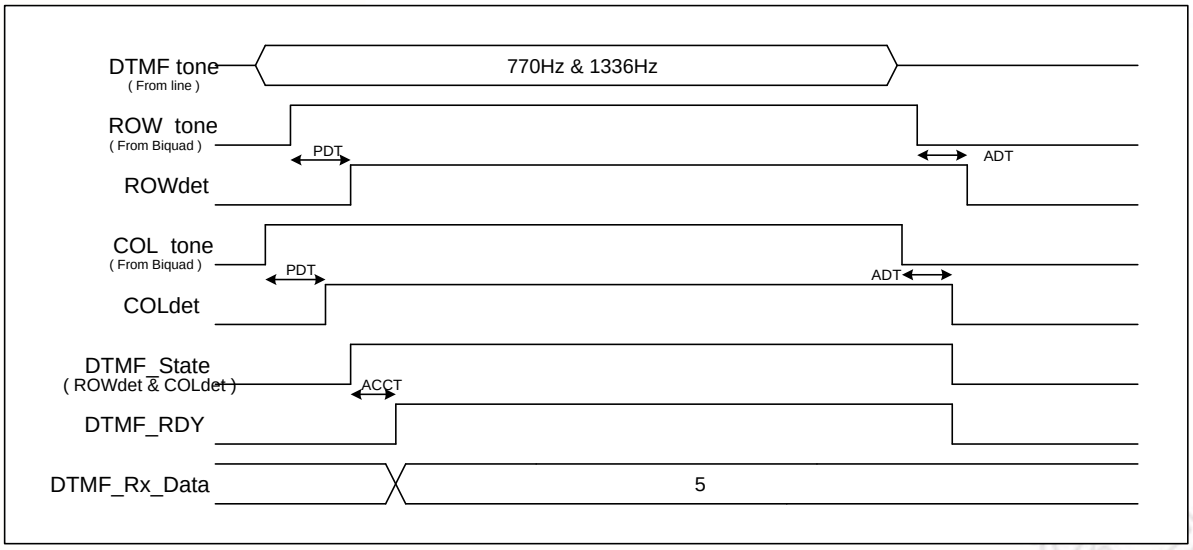


Figure 8-13 DTMF Detector Acquisition Timing

Table 8-10 DTMF Tone Decoding

		Column frequency			
Row Frequency		1209 Hz	1336 Hz	1477 Hz	1633 Hz
	697 Hz	0x01 hex	0x02 hex	0x03 hex	0x0D hex
		1	2	3	A
	770 Hz	0x04 hex	0x05 hex	0x06 hex	0x0E hex
		4	5	6	B
	852 Hz	0x07 hex	0x08 hex	0x09 hex	0x0F hex
		7	8	9	C
	941 Hz	0x0B hex	0x0A hex	0x0C hex	0x00 hex
		*	0	#	D

A detailed description of the DTMF Decoder control registers follows:

8.10.1 DTMF_CTRL

Address	Access Mode	Value At Reset	Nominal Value
0x1E0	R/W	0x00	

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
DTMF_EN	Reserved	DTMF_FREQ_DEV		DTMF_TC			

DTMF_TC — Sets the time constant used for DTMF signal energy level estimation. The DTMF_TC constant controls the speed at which the energy converges. Smaller values of DTMF_TC cause faster convergence at the expense of lower precision. Higher values of DTMF_TC will provide a more precise estimate of the energy while trading off a slower convergence time.

DTMF_FREQ_DEV — The allowable frequency deviation of DTMF tone detector. As given in Table 8-11

DTMF_EN — Enable DTMF detection block.

Table 8-11 DTMF Frequency Deviation Register

DTMF_FREQ_DEV	Frequency Deviation
0	2.0%
1	2.5%
2	3.0%
3	3.5%

8.10.2 DTMF_FIFO_CTRL

Address	Access Mode	Value At Reset	Nominal Value
0x1E1	R/W	0x40	

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Reserved	FIFO_LEN			Reserved	Reserved	Reserved	FIFO_CLR

FIFO_LEN — Sets the FIFO level at which the DTMF detector generates an interrupt to the host. Interrupt is generated when FIFO_LEN+1 tones have been received.

FIFO_CLR — Resets FIFO to empty and clear the content of the FIFO.

8.10.3 DTMF_FIFO

Address	Access Mode	Value At Reset	Nominal Value
0x1E2	R	0x00	

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
FIFO_DEPTH				FIFO_DATA			

FIFO_DATA A read to this register reads back the head of the FIFO and **increments read pointer to next address**.

FIFO_DEPTH Is the number of entries remaining in the FIFO **before** the current read operation.

8.10.4 DTMF_FIFO_STATUS

Address	Access Mode	Value At Reset	Nominal Value
0x1E3	R/W	0x01	

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
FIFO_DEPTH				Reserved	Reserved	FIFO_FULL	FIFO_EMP

FIFO_DEPTH Indicates the number of entries present in the DTMF FIFO, same value as FIFO_DEPTH of previous register

FIFO_FULL Read only active high bit indicating when FIFO is full. Any data received while this bit is high is lost. Total depth of the DTMF FIFO is 8.

FIFO_EMP Read only active high bit indicating when FIFO is empty.

8.10.5 DTMF_THRES

Address	Access Mode	Value At Reset	Nominal Value
0x1E4~0x1E5	R/W	0x0100	

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
E4	DTMF_THRES[15:8]							
E5	DTMF_THRES[7:0]							

DTMF_THRES DTMF Threshold, signal level required to qualify as a valid DTMF signal. The threshold in dB is given by 10 * $\log_{10}$ (DTMF_THRES * 2^{-22}). The DTMF threshold ranges from -12 dB to -66 dB, and the default is -42 dB. If the signal level is below DTMF_THRES, the results of the frequency estimator are ignored and the signal will not be detected as a valid DTMF signal.

8.10.6 DTMF_PDT

Address	Access Mode	Value At Reset	Nominal Value
0x1E6	R/W	0x00	

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
DTMF_PDT							

DTMF_PDT	This is the Present Detect Time, the time for which a tone must be present to be qualified as a valid DTMF tone. Units are 0.5ms, range is 0-127.5ms.

新唐科技 nuvoTon INTELLECTUAL PROPERTY

8.10.7 DTMF_ADT

Address	Access Mode	Value At Reset	Nominal Value
0x1E7	R/W	0x00	

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
DTMF_ADT							

DTMF_ADT — This is the Absent Detect Time, the time for which a tone must be absent before a signal is considered a new DTMF tone. Units are 0.5ms, range is 0-127.5ms.

8.10.8 DTMF_ACCT

Address	Access Mode	Value At Reset	Nominal Value
0x1E8	R/W	0x00	

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
DTMF_ACCT							

DTMF_ACCT — This is the Accept Time, (DTMF_ACCT + 1) * 0.5ms, the time for which a tone must be stable to be qualified as a correct tone. Units are 0.5ms, range is 0.5-128ms. This guard time improves detection performance by rejecting detected signals with insufficient duration and by masking momentary detection dropout.

8.10.9

8.10.10DTMF_RX_DATA

Address	Access Mode	Value At Reset	Nominal Value
0x1EA	R	0x00	

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
DTMF_RDY	DTMF_STATE	RESERVED	RESERVED	DTMF_RX_DATA			

DTMF_RX_DATA — DTMF Detector received data. This register can be used as an alternative to the FIFO. This data is valid when DTMF_RDY is active.

DTMF_STATE — Indicates whether a valid DTMF tone is currently being detected and present time (PDT) is qualified.

DTMF_RDY — Indicates that a valid DTMF tone has been present for required hold time (ACCT).

8.10.11DTMF_ROW_FREQ

Address	Access Mode	Value At Reset	Nominal Value
0x1EB-0x1EC	R	0x0000	

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
EB	DTMF Row Frequency [15:8]							
EC	DTMF Row Frequency [7:0]							

DTMF Row Frequency	DTMF_ROW_FREQ[15:3] is the integer part of the DTMF Row frequency, DTMF_ROW_FREQ[2:0] is the decimal fraction part of the DTMF Row frequency (13.3 format). These two bytes are for debug mode, and display the DTMF Row frequency directly.

8.10.12DTMF_COL_FREQ

Address	Access Mode	Value At Reset	Nominal Value
0x1ED-0x1EE	R	0x0000	

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ED	DTMF Column Frequency[15:8]							
EE	DTMF Column Frequency[7:0]							

DTMF Column Frequency	DTMF_COL_FREQ[15:3] is the integer part of the DTMF Column frequency, DTMF_COL_FREQ[2:0] is the decimal fraction part of the DTMF Column frequency (13.3 format). These two bytes are for debug mode, and display the DTMF Column frequency directly.

8.10.13Tip & Tricks

Below is an example of an interrupt service routine for the DTMF detector interrupt.

```
unsigned char ucVal;

ucVal = ReadRegister ( CFG_DTMF_FIFO_STATUS );
if( !(ucVal & 0x01)){
      do{
            // FIFO not empty - get it
            ucVal = ReadRegister ( CFG_DTMF_FIFO );
            // Store data in a FIFO for processing
            dtmfFIFO.data[dtmfFIFO.w_ptr++] = ucVal & 0x0F;
      } while( ((ucVal>>4) & 0xf)>1 );
}
```

8.11 DTMF and Arbitrary Tone Generation.

The ISD61S00 has two independently programmable voice-band oscillators for DTMF and FSK tone generation. The tone generators can be controlled by individual registers or automatically by the DTMF or FSK generation sub-systems.

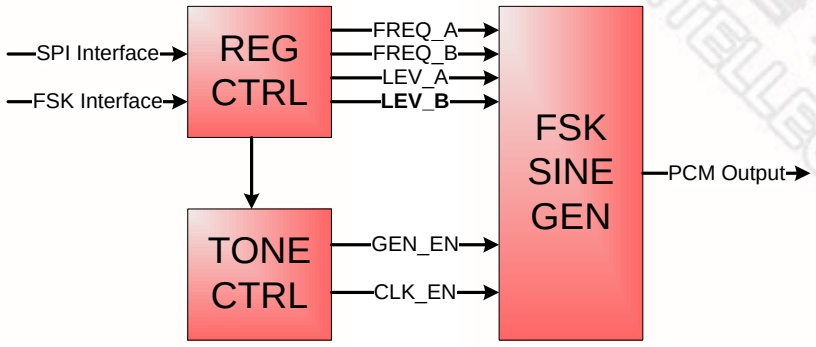


Figure 8-14 Tone Generator Block Diagram

The tone generator block operates in three different modes:

1. DTMF Mode: in this mode DTMF digits are generated from data read from a FIFO for a register-specified on and off time time.
2. Programmable Mode: This mode generates a user-defined tone for a register-specified on and time time.
3. Continuous Mode: This mode generates a user-defined tone continuously until the user intervenes to stop generation.

In Dual Tone Multi Frequency (DTMF) mode two tones are used to generate a DTMF digit. One tone is chosen from four possible row tones, while the other tone is chosen from four possible column tones. The sum of these tones constitutes one of 16 possible DTMF digits as shown in Table 8-13. Tones are generated according to the data written in the TONE_INDEX_X registers. Up to twenty four DTMF digits can be loaded into these registers for transmission. The register TONE_LENGTH determines how many of these DTMF digits will be generated.

For example, to generate the DTMF string 4085442222, TONE_LENGTH is set to 10, the number of digits to transmit, TONE_INDEX_0 is loaded with 0x40, TONE_INDEX_1 is loaded with 0x85 etc.. To generate the tones, DTMF_MODE, GEN_EN and CLK_EN are all set high in the TONE_CTRL register. When digits have been generated GEN_EN will be self-cleared indicating operation has finished.

In programmable mode the TONE_LENGTH register sets the number of tones to generate. Tones have a duration of TONE_ON_TIME and are spaced TONE_OFF_TIME apart. To generate a single tone of longer than 255ms the TONE_OFF_TIME can be set to zero and a tone of TONE_LENGTH*TONE_ON_TIME ms can be generated.

Continuous mode is entered by setting TONE_ON_TIME, TONE_OFF_TIME, and TONE_LENGTH all to 0. The user may then set the frequency and level for tones A and/or B, which will then be generated continuously as long as GEN_EN and CLK_EN are set high.

Configuration for the three modes is listed below.

Table 8-12 Tone Generation Mode Setup

Mode	TONE_FREQ_A TONE_FREQ_B	TONE_LEVEL_A TONE_LEVEL_B	TONE_ON_TIME	TONE_OFF_TIME	TONE_LENGTH	INDEX_n

Continuous Mode	User defined	User defined	Set : 0	Set : 0	Set : 0	Don't care
DTMF Mode	Don't care	User defined	User defined	User defined	Number of digits in tone index registers.	Digits to send
Programmable Mode	User defined	User defined	User defined	User defined	Number of tones to generate.	Don't care

8.11.1 TONE_CTRL

Address	Access Mode	Value At Reset	Nominal Value
0x1C0	R/W	0x00	

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CLK_EN	Reserved	Reserved	Reserved	Reserved	Reserved	DTMF_MODE	GEN_EN

GEN_EN 1: Start tone generation. 0: Write a zero to stop tone generation. GEN_EN will self-clear after completion of tone generation in DTMF mode or in programmable mode.

DTMF_MODE 1: DTMF mode. 0: Programmable tone mode.

CLK_EN 1: Enable clk. 0: Disable clk. To minimize power consumed by the tone generator, clock should be disabled when not in use.

8.11.2 TONE_FREQ_A

Address	Access Mode	Value At Reset	Nominal Value
0x1C1-0x1C2	R/W	0x0000	

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
C1	Reserved	TONE_FREQ_A[14:8]						
C2	TONE_FREQ_A[7:0]							

TONE_FREQ_A Sets the frequency of tone A to TONE_FREQ_A * 2.5 Hz. If set to zero, tone A is disabled. Register is ignored in DTMF mode. Valid range for this register is 0x0000 – 0x0640 (4 kHz).

8.11.3 TONE_FREQ_B

Address	Access Mode	Value At Reset	Nominal Value
0x1C3-0x1C4	R/W	0x0000	

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
C3	Reserved	TONE_FREQ_B[14:8]						
C4	TONE_FREQ_B[7:0]							

TONE_FREQ_B Sets the frequency of tone B to TONE_FREQ_B * 2.5 Hz. If set to zero, tone B is disabled. Register is ignored in DTMF mode. Valid range for this register is 0x0000 – 0x0640 (4 kHz).

8.11.4 TONE_LEVEL_A

Address	Access Mode	Value At Reset	Nominal Value
0x1C5	R/W	0x00	

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
		TONE_LEV_A					

TONE_LEV_A Sets the level of tone A relative to full scale. Level (dBFS) = (TONE_LEV_A-63)/2

8.11.5 TONE_LEVEL_B

Address	Access Mode	Value At Reset	Nominal Value
0x1C6	R/W	0x00	

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
		TONE_LEV_B					

TONE_LEV_A Sets the level of tone A relative to full scale. Level (dBFS) = (TONE_LEV_A-63)/2

8.11.6 TONE_ON_TIME

Address	Access Mode	Value At Reset	Nominal Value
0x1C7	R/W	0x00	

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TONE_ON_TIME							

TONE_ON_TIME Sets the present (on) time of each tone to TONE_ON_TIME * 1ms. Set to zero for continuous mode.

8.11.7 TONE_OFF_TIME

Address	Access Mode	Value At Reset	Nominal Value
0x1C8	R/W	0x00	

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TONE_OFF_TIME							

TONE_OFF_TIME It sets the absent (off) time between each tone to TONE_OFF_TIME * 1ms. Set to zero for continuous mode.

8.11.8 TONE_LENGTH

Address	Access Mode	Value At Reset	Nominal Value
0x1C9	R/W	0x00	

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Reserved	Reserved	Reserved	TONE_LENGTH				

TONE_LENGTH This register determines the number of DTMF digits to be sent in DTMF mode. Digits are stored, two per byte, in the TONE_INDEX_X registers. The maximum number of DTMF digits is 24.

In programmable tone mode, the tone length sets the number of tones to generate, maximum number is 24.

For continuous mode, TONE_LENGTH must be set to 0.

8.11.9

8.11.10TONE_INDEX_0 ~ TONE_INDEX_A

Address	Access Mode	Value At Reset	Nominal Value
0x1D0-0x1DB	R/W	0x00	

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
D0	Digit 0				Digit 1			
D1	Digit 2				Digit 3			
D2	Digit 4				Digit 5			
D3	Digit 6				Digit 7			
D4	Digit 8				Digit 9			
D5	Digit 10				Digit 11			
D6	Digit 12				Digit 13			
D7	Digit 14				Digit 15			
D8	Digit 16				Digit 17			
D9	Digit 18				Digit 19			
DA	Digit 20				Digit 21			
DB	Digit 22				Digit 23			

Digit 0 ~ 23 These registers can be loaded with up to 24 DTMF digits. The DTMF generator will send the fall TONE_LENGTH digits in this register bank sequentially. The frequencies generated by each nibble are defined in Table 8-13.

Table 8-13 DTMF Frequency Mapping.

		Column frequency			
Row Frequency		**1209 Hz**	**1336 Hz**	**1477 Hz**	**1633 Hz**
	697 Hz	0x01 hex	0x02 hex	0x03 hex	0x0D hex
		1	2	3	A
	770 Hz	0x04 hex	0x05 hex	0x06 hex	0x0E hex
		4	5	6	B
	852 Hz	0x07 hex	0x08 hex	0x09 hex	0x0F hex
		7	8	9	C
	941 Hz	0x0B hex	0x0A hex	0x0C hex	0x00 hex
		*	0	#	D

8.12 FSK Generation

The ISD61XX provides an FSK generator supporting Bell 202, Bell 103, ITU-T V.23 and ITU-T V.21 standard FSK. The FSK Generator controls the tone generator to generate phase continuous tones at the appropriate frequency and baud rate. The generator can be configured to package bytes with selectable start and stop bits and a parity bit. Alternatively, the user may enable software mode (SW_MODE) and configure mark and space frequencies and baud rate generate arbitrary FSK waveforms.

The block diagram of the FSK generator is shown below. Data to be transmitted is loaded into the 8 deep FIFO. The interface can be configured to generate interrupts to the host when this FIFO is empty or less than half full. The protocol for transmission is selected by the control registers FSKE_CTRL1 and FSK_CTRL2. Bits in these registers allow the user to configure the baud rate, mark and space frequencies, parity bit generation and START and STOP bit generation.

FSK transmission is initiated by asserting FSK_EN bit. The FSK transmit data is clocked serially out of the FIFO one byte at a time beginning with the LSB. If package mode (PACKAGE_EN) is enabled, a 'start bit' (Space) will automatically be added to the head of the FSK transmit data. Furthermore, one or two 'stop bits' (Mark) are added to the end of the FSK transmit data, depending on the setting of STOP_SEL. If package mode is not enabled the FSK transmit data is transmitted as it appears in the FSK FIFO.

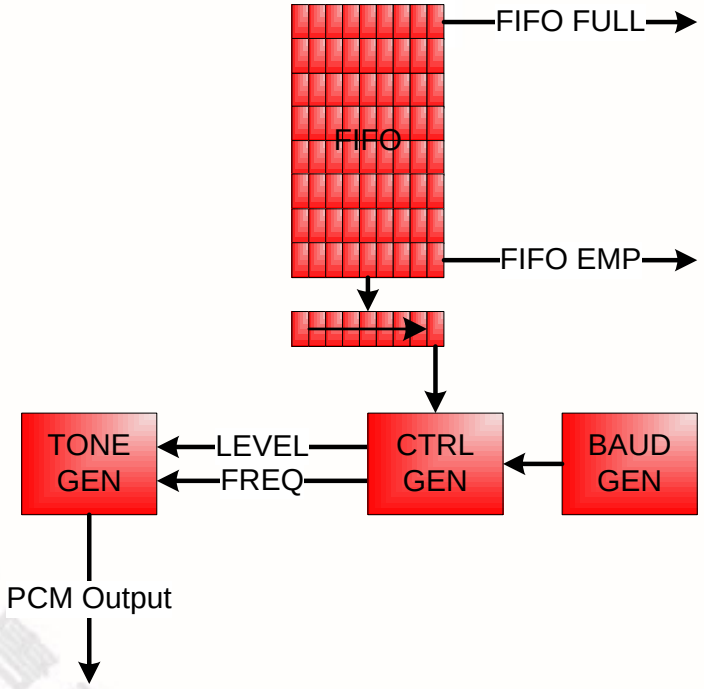


Figure 8-15 Architecture of Linear FSK Waveform Generator

8.12.1 FSKE_CTRL1

Address	Access Mode	Value At Reset	Nominal Value
0x160	R/W	0x00	

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
FSK_EN	PACKAGE_EN	HALF_FIFO_INT	FSK_TX_ON	STOP_SEL	RESERVED		

FSK_EN 1: Enable FSK encoder. 0: Power down FSK encoder.

PACKAGE_EN 1: Package mode. 0: Normal mode. In package mode data byte will be packaged as follows:

- A start bit at SPACE frequency.
- The data byte.
- A parity bit as determined by FSKE_CTRL2
- One or two stop bits at MARK frequency depending on STOP_SEL bit.

HALF_FIFO_INT Enables interrupt when FIFO depth changes from 5 to 4 (half empty condition). The FIFO also generates an interrupt when FSK_TX_ON is enabled and the FIFO is empty.

FSK_TX_ON When set FSK Encoder will transmit data from transmit shift register. Also when set encoder will generate an interrupt when FIFO is empty.

STOP_SEL 1: Two STOP bits generated. 0: One STOP bit generated.

8.12.2 FSKE_CTRL2

Address	Access Mode	Value At Reset	Nominal Value
0x161	R/W	0x23	

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PARITY_EN	PARITY_TYPE	IIR_EN	RESERVED	BAUD_110	SPEC_SEL		

PARITY_EN 1: Enable parity generation. 0: Disable parity generation.

PARITY_TYPE 1: Even parity. 0: Odd parity.

IIR_EN Enable IIR filter for FSK power mask specification. (Default on)

BAUD_110 When set generate FSK at 110 baud if Bell 103 selected.

SPEC_SEL Determines the FSK specification that is generated according to Table 8-14.

Table 8-14 FSK Encoder Specification Selection.

SPEC_SEL	000	001	010	011	100	101	110	111
Spec.	V.23	V.23	Bell 202	Bell 202	Bell 103	Bell 103	V.21	V.21

Baud Rate	0	75 bps	1200 bps	150 bps	1200 bps	300 bps	300 bps	300 bps	300 bps
	1					110 bps	110 bps		
Mark '1'		390Hz	1300Hz	387Hz	1200Hz	1270Hz	2225Hz	980Hz	1650Hz
Space '0'		450Hz	2100Hz	487Hz	2200Hz	1070Hz	2025Hz	1180Hz	1850Hz

8.12.3 FSKE_TX_DATA

Address	Access Mode	Value At Reset	Nominal Value
0x162	W	0x00	

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
FSK_TX_DATA							

FSK_TX_DATA A write to this register pushes the data onto the Tx FIFO.

8.12.4 FSKE_STATUS

Address	Access Mode	Value At Reset	Nominal Value
0x163	R	0x00	

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
RESERVED	RESERVED	FIFO_EMP	FIFO_FULL	FIFO_DEPTH			

FIFO_DEPTH Returns the current number of entries in the Tx FIFO.

FIFO_FULL Set when Tx FIFO is full.

FIFO_EMP Set when Tx FIFO is empty.

8.12.5 FSKE_GAIN

Address	Access Mode	Value At Reset	Nominal Value
0x164	R/W	0x39	

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
RESERVED	RESERVED	FSK_GAIN					

FSK_GAIN This register controls the gain of the generated FSK waveform relative to full scale. Gain (dBFS) = (FSK_GAIN-63)/2.0 Gain range is 0 to -31.5dBFS

8.12.6 FSKE_PROG

Address	Access Mode	Value At Reset	Nominal Value
0x165	R/W	0x00	

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

Address	Access Mode	Value At Reset	Nominal Value

SW_MODE	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	SW_FLT

SW_MODE When set, SPEC_SEL is ignored and the FSK encoder frequencies and baud rate are controlled by FSKE_MARK_FREQ, FSKE_SPACE_FREQ, and FSKE_BAUD described below.

SW_FLT This register controls the specification for the filter controlled by IIR_EN (0x23.5)

0: use built in filter coefficients.

1: use software programmable filter coefficients. These coefficients are stored in the CAS/ATD tone detector RAM accessible through registers 0x274~0x275

8.12.7 FSKE_BAUD (SW_MODE = 1)

Address	Access Mode	Value At Reset	Nominal Value
0x166~0x168	R/W	0x00000	

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	BAUD_RATE_CNT[17:16]	
BAUD_RATE_CNT[15:8]							
BAUD_RATE_CNT[7:0]							

BAUD_RATE_CNT In SW_MODE these registers determine the baud rate of generated FSK. Each bit is generated for BAUD_RATE_CNT * 81.38ns.

Baud rate = 1/(BAUD_RATE_CNT * 81.38e-9)

BAUD_RATE_CNT = 12,288,000/baud rate

8.12.8 FSKE_MARK_FREQ (SW_MODE = 1)

Address	Access Mode	Value At Reset	Nominal Value
0x169~0x16A	R/W	0x00	

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
RESERVED	MARK_FREQ[14:8]						
MARK_FREQ[7:0]							

MARK_FREQ In SW_MODE these registers determine the mark frequency of the generated FSK. Frequency is given by MARK_FREQ * 2.5Hz.

8.12.9 FSKE_SPACE_FREQ (SW_MODE = 1)

Address	Access Mode	Value At Reset	Nominal Value
0x16B~0x16C	R/W	0x00	

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
RESERVED	SPACE_FREQ[14:8]						
SPACE_FREQ[7:0]							

SPACE_FREQ In SW_MODE these registers determine the space frequency of the generated FSK. Frequency is given by SPACE_FREQ * 2.5Hz.

8.12.10 FSKE_TEST

Address	Access Mode	Value At Reset	Nominal Value
0x16D	R/W	0x00	

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
RESERVED	RESERVED	RESERVED	RESERVED	AUTO_GEN	RESERVED	RESERVED	RESERVED

AUTO_GEN Setting AUTO_GEN will implicitly set FSK_EN, FSK_TX_ON, PACKAGE_EN, and STOP_SEL. The FSK_FIFO will be fed random data generated by LFSR.

8.12.11

8.12.12 Example FSK Generator Usage

Below is an example of an interrupt service state machine to service the FSK generation interrupt. The general configuration of the encoder and path is assumed to be setup and a 10ms timer tick interrupt also enabled. The routine will send out carrier synchronization signal along with data loaded into a transmit FIFO.

```
PUBLIC void vFSKGenFSM(void)
{

        static byte byCounter;
        byte i, tmp;
```

```
byte bNumChanSeizeBytes = NUM_SEIZE;
byte bNumMarkBytes = NUM_MARK;

switch(byFSKGenState)
{
        case FSKG_IDLE:
        if(bUsbCtrl[0] & USB_CTRL0_FSK_GEN_START)
        {
                byFSKGenState = FSKG_CHANNEL_SEIZURE;
        }
        break;
        case FSKG_CHANNEL_SEIZURE:
        // The assumption here is that the FSK encoder block
        // general configuration has been setup already now
        // we just turn off package mode and send channel seizure
        // signal.
        tmp = ReadRegister(CFG_FSKE_CTRL1);
        tmp &= ~CFG_FSKE_CTRL1__PACKAGE_EN;
        tmp |= CFG_FSKE_CTRL1__FSK_TX_ON | CFG_FSKE_CTRL1__FSK_EN
        |CFG_FSKE_CTRL1__HALF_FIFO_INT;
        writeCfgReg(CFG_FSKE_CTRL1,tmp);
        tmp = ReadRegister(CFG_FSKE_CTRL2);
        tmp &= ~CFG_FSKE_CTRL2__PARITY_EN;
        writeCfgReg(CFG_FSKE_CTRL2,tmp);
        if(bNumChanSeizeBytes > 8)
        {
                // Load first 8 bytes into FSK FIFO
                for (i=0;i<8;i++)
                {
                        writeCfgReg(CFG_FSKE_FIFO_W,0x55);
                }
                byFSKGenState = FSKG_CHANNEL_SEIZURE2;
                bNumChanSeizeBytes -=8;
        }
        else if(bNumChanSeizeBytes != 0)
        {
        // Load  bytes into FSK FIFO
                for (i=0;i< bNumChanSeizeBytes;i++)
                {
                        writeCfgReg(CFG_FSKE_FIFO_W,0x55);
                }
                byFSKGenState = FSKG_CARRIER_GEN;
                bNumChanSeizeBytes = 0;
        }
        else
        {
                // No Channel Seizure. Add MARK and go to carrier gen.
                writeCfgReg(CFG_FSKE_FIFO_W,0xFF);
                byFSKGenState = FSKG_CARRIER_GEN;
        }
```

```
byCounter = 0;
break;
case FSKG_CHANNEL_SEIZURE2:
if(DevStatus[2] & STATUS2_FSK_E)
{
        // We get here FIFO interrupt has occurred. There is space for
        // at least 4 bytes (half interrupt set)
        if(bNumChanSeizeBytes > 3)
        {
        // Load 4 bytes into FSK FIFO
                for (i=0;i<4;i++)
                {
                        writeCfgReg(CFG_FSKE_FIFO_W,0x55);
                }
                byFSKGenState = FSKG_CHANNEL_SEIZURE2;
                bNumChanSeizeBytes -=4;
        }
        else
        {
        // Load  bytes into FSK FIFO
                for (i=0;i< bNumChanSeizeBytes;i++)
                {
                        writeCfgReg(CFG_FSKE_FIFO_W,0x55);
                }
                writeCfgReg(CFG_FSKE_FIFO_W,0xFF);
                byFSKGenState = FSKG_CARRIER_GEN;
                bNumChanSeizeBytes = 0;
        }
}
break;
case FSKG_CARRIER_GEN:
// Getting here means we have finished loading sych signal
// but does not mean that FIFO is empty. Next we need to wait
// a certain number of ms before sending first data.This can be
// achieved by sending 0xFF.
if(DevStatus[2] & STATUS2_FSK_E)
{
        // We get here FIFO interrupt has occurred. There is space for
        // at least 4 bytes (half interrupt set)
        if(bNumMarkBytes > 3)
        {
        // Load 4 bytes into FSK FIFO
                for (i=0;i<4;i++)
                {
                        writeCfgReg(CFG_FSKE_FIFO_W,0xFF);
                }
                byFSKGenState = FSKG_CARRIER_GEN;
                bNumMarkBytes -=4;
        }
        else
        {
```

```
                        // Load  bytes into FSK FIFO
                                for (i=0;i< bNumMarkBytes;i++)
                                {
                                        writeCfgReg(CFG_FSKE_FIFO_W,0xFF);
                                }
                                writeCfgReg(CFG_FSKE_FIFO_W,0xFF);

                                // Ensure that next INT is for FIFO empty
                                tmp = ReadRegister(CFG_FSKE_CTRL1);
                                tmp &= ~CFG_FSKE_CTRL1__HALF_FIFO_INT;
                                writeCfgReg(CFG_FSKE_CTRL1,tmp);
                                byCounter = 0;
                                byFSKGenState = FSKG_DATA_CONFIG;
                                bNumMarkBytes = 0;
                        }
                }
                break;
                case FSKG_DATA_CONFIG:
                if(DevStatus[2] & STATUS2_FSK_E)
                {
                // Carrier phase finished set up data transfer.
                        tmp = ReadRegister(CFG_FSKE_CTRL1);
                        tmp |= CFG_FSKE_CTRL1__PACKAGE_EN | CFG_FSKE_CTRL1__FSK_TX_ON |
CFG_FSKE_CTRL1__FSK_EN | CFG_FSKE_CTRL1__HALF_FIFO_INT;
                        writeCfgReg(CFG_FSKE_CTRL1,tmp);
                        tmp = ReadRegister(CFG_FSKE_CTRL2);
                        tmp |= CFG_FSKE_CTRL2__PARITY_EN;
                        writeCfgReg(CFG_FSKE_CTRL2,tmp);

                        if(TxDataFifo.Depth >= 8)
                        {
                                // Load the first 8 samples
                                for (i=0;i<8;i++)
                                {

      writeCfgReg(CFG_FSKE_FIFO_W,TxDataFifo.Fifo_Buffer[(TxDataFifo.ReadIndex++ & 0x3F)]);
                                }
                                TxDataFifo.Depth -= 8;
                        }
                        byFSKGenState = FSKG_DATA_TX;
                }
                break;
                case FSKG_DATA_TX:
                if((DevStatus[2] & STATUS2_FSK_E) || (DevStatus[3] & STATUS3_TIMER_INT))
                {
                        // Get here either for a FIFO int or TIMER interrupt.
                        // Find amount of space on FIFO
                        tmp = 8 - (ReadRegister(CFG_FSKE_STATUS) & 0x0F);
                        if(TxDataFifo.Depth >= tmp)
                        {
                                // There are tmp available bytes to transmit
```

```
                        for (i=0;i<tmp;i++)
                        {
writeCfgReg(CFG_FSKE_FIFO_W,TxDataFifo.Fifo_Buffer[(TxDataFifo.ReadIndex++ & 0x3F)]);
                        }
                        TxDataFifo.Depth -= tmp;
                }
                else if(TxDataFifo.Depth)
                {
                        // There are more bytes to Tx
                        for (i=0;i<TxDataFifo.Depth;i++)
                        {
writeCfgReg(CFG_FSKE_FIFO_W,TxDataFifo.Fifo_Buffer[(TxDataFifo.ReadIndex++ & 0x3F)]);
                        }
                        TxDataFifo.Depth -= TxDataFifo.Depth;
                }
                else
                {
                        // Only get to here if TxDataFIFO is empty. Check for EOT
                        // condition which is USB_CTRL0_FSK_GEN_START set low. If so
                        // and FSK FIFO is empty we can end transmission.
                        if(((bUsbCtrl[0] & USB_CTRL0_FSK_GEN_START)==0)
                                && (tmp == 8))
                        {
                                tmp = ReadRegister(CFG_FSKE_CTRL1);
                                tmp &= ~(CFG_FSKE_CTRL1__FSK_TX_ON |CFG_FSKE_CTRL1__FSK_EN);
                                writeCfgReg(CFG_FSKE_CTRL1,tmp);
                                byFSKGenState = FSKG_IDLE;
                        }
                }
        }

        break;
        default:
        byFSKGenState = FSKG_IDLE;
        break;
}

}
```

8.13 FSK Detection

The ISD61XX has an FSK Detector supporting Bell 202, Bell 103, ITU-T V.21 and ITU-T V.23 standard FSK. The detector is capable of operating at baud rates of 75, 110, 150, 300, and 1200bps. The architecture of the FSK detection system is shown in Figure 8-16 Block Diagram of FSK Detector

.The system consists of filtering the incoming signal followed by a match filter to extract FSK data. Additional blocks perform carrier detection and clock recovery to synchronize the data and load it into a FIFO for host processing. The match filter consists of a shift register sampling the signal at 48kHz. This sampled waveform is compared to the matching templates impulse_L and impulse_H for the mark and space frequencies each sample period. A peak detector determines which of the frequencies is present and outputs the result.

8-bit FSK data is packaged along with optional START, STOP and PARITY bits. The expected format of the data is configurable. Transmission of a '1' bit is called a MARK, transmission of a '0' bit is called a SPACE. MARK and SPACE frequencies are determined by the FSK standard configured. The 8-bit data is transmitted LSB first.

The match filter calculates a matching score for the SPACE (Y0) and MARK (Y1) for each sample period. The score is determined by rotating the pattern template against the input signal (FSK_IN) and counting the number of matches to the template giving a score between 0 (no matching bits) and N (maximal matching bits). To improve detection accuracy and noise immunity, the detection can be performed over multiple samples. This is controlled by the FSKD_JUDGE register where detection can operate between one and ten sample periods.

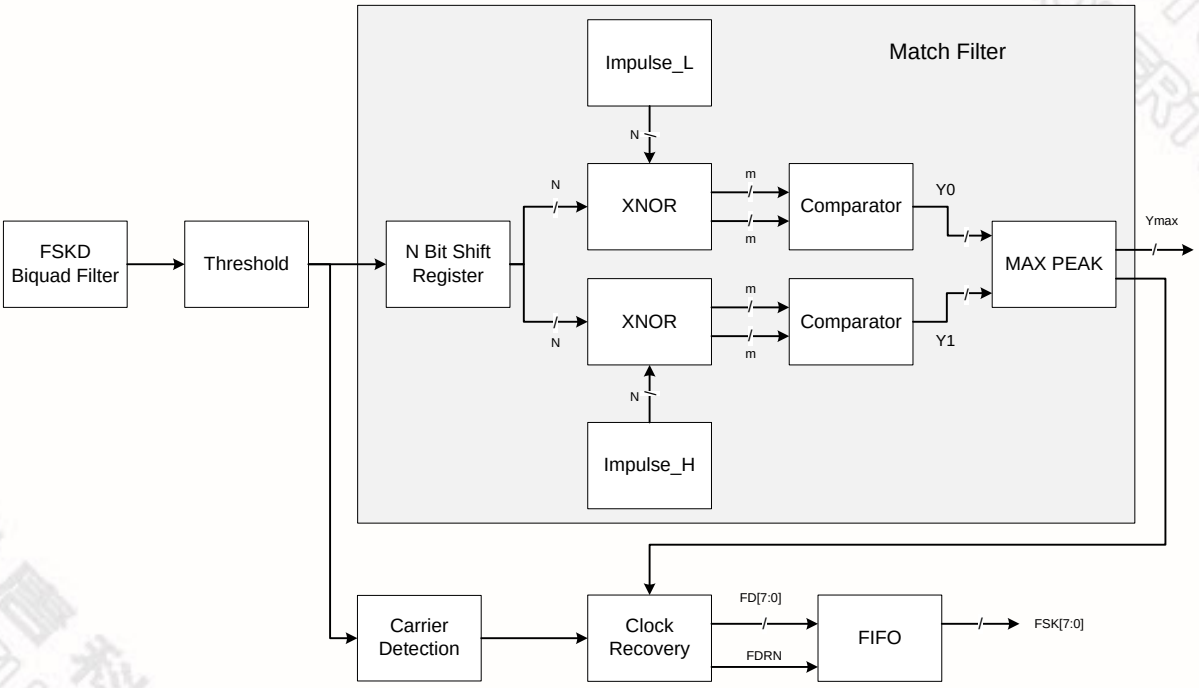


Figure 8-16 Block Diagram of FSK Detector

8.13.1 FSKD_CTRL

Address	Access Mode	Value At Reset	Nominal Value
0x180	R/W	0x00	

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

Address	Access Mode	Value At Reset	Nominal Value

FSK_EN	CDB_DET_MD	MARK_DET	RESERVED	RESERVED	RESERVED	RESERVED	FSK_INT_MD

FSK_INT_MD	1: The FSK interrupt will only occur when a 0x55 or 0xAA byte is detected; this mode is used for detecting channel seizure signal. 0: The FSK interrupt will occur normally; this mode is used for receiving data transmission. The FSK interrupt will occur based on the FIFO_LEN parameter of register FSKD_FIFO_CTRL. The interrupt will not occur again until FSKD_FIFO_DOUT register is read.
MARK_DET	0: Normal operation. 1: For type II CID only when FSK session does not contain channel seizure. Causes carrier detection to only look for MARK signal when in Bell 202 or V.23 mode.
CDB_DET_MD	0: Type 1 FSK detect. Channel seizure will be detected for CDB. 1: Type 2 FSK detect. Mark and space frequency will be detected for CDB.
FSK_EN	1: Enable FSK detection block. 0: FSK detection block powered down.

Two types of FSK sessions are common. Type 1 session begins with a channel seizure phase where alternate mark and space signals are transmitted to allow channel synchronization. Type 2 FSK has no channel seizure phase but rather begins with long MARK signal. To trigger operation of the FSK detector, the circuit first determines whether an FSK carrier is present. This is reflected by the CDB (carrier detect bar) bit in FSKD_STATUS. Once a carrier is detected the FSK demodulator is enabled and data is decoded. The carrier detect condition is determined by the following table:

CDB_DET_MD	MARK_DET	Bell103, V.21	Bell202,V.23
0	0	Detect FSK channel seizure and in-band mark or space frequency. Use for Type 1 FSK.	Detect FSK channel seizure and in-band energy level.* Use for Type 1 FSK.
1	0	Detect FSK in-band mark or space frequency.† Use for Type 2 FSK.	Detect FSK in-band mark or space frequency.† Use for Type 2 FSK.
1	1	Detect FSK in-band mark or space frequency.‡ Use for Type 2 FSK	Detect FSK in-band mark frequency only.

* For Bell202 and V.23, circuit will judge in-band energy level specified by FSKD_ENERGY_HIGH_TH (0x18C) and FSKD_ENERGY_LOW_TH (0x18D) registers for the CDB on and off condition.

† When FSK doesn't contain channel seizure and starts with long mark, This mode only detect in-band mark and space frequency, software should check FSKD_STATUS (0x183) register FSK_DF_OUT bit for determining mark duration.

‡ When FSK doesn't contain channel seizure and starts with long mark, This mode only detect in-band mark and space frequency, software should check FSKD_STATUS (0x183) register FSK_DF_OUT bit for determining mark duration.

8.13.2 FSKD_MODE

Address	Access Mode	Value At Reset	Nominal Value
0x181	R/W	0x04	

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Reserved	Reserved	PARITY_EN	PARITY_TYPE	MODE[3:0]			

PARITY_EN When set decoder expects a parity bit in the received data following the 8 data bits. Parity error status is reported in bit 3 of FSKD_FIFO_STATUS (0x18B).

PARITY_TYPE 1: even parity, 0: odd parity

MODE[3:0] Selects the mode of FSK detection. Valid mode settings are shown in Table 8-15.

FSKMode[3:0]	0111	0110	0101	0100	0011	0010	0001	0000
Spec.	V.21	V.21	Bell 103	Bell 103	Bell 202		V.23	
Baud Rate	300 bps	300 bps	300 bps	300 bps	1200 bps		1200 bps	
Mark '1'	1650	980Hz	2225Hz	1270Hz	1200Hz		1300Hz	
Space '0'	1850	1180Hz	2025Hz	1070Hz	2200Hz		2100Hz	

FSKMode[3:0]	1111	1110	1101	1100	1011	1010	1001	1000
Spec.	Bell 103	Bell 103	Bell 103	Bell 103	Bell 103	Bell 103		
Baud Rate	75 bps	75 bps	110 bps	110 bps	150 bps	150 bps		
Mark '1'	2225Hz	1270Hz	2225Hz	1270Hz	2225Hz	1270Hz		
Space '0'	2025Hz	1070Hz	2025Hz	1070Hz	2025Hz	1070Hz		

Table 8-15 FSK Detector Modes.

8.13.3 FSKD_ADJUST

Address	Access Mode	Value At Reset	Nominal Value
0x182	R/W	0xC0	

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DEC_MD_SEL	SYM_ADJ_MD	CF_ADJ_SW	RESERVED	CORR_ADJ_S	CORR_ADJ[2:0]		

CORR_ADJ[2:0]* Match filter space frequency correlation adjust value.

* CORR_ADJ_S and CORR_ADJ[2:0] are used to micro tune internal match filter space correlation value. This is used to compensate systematic mark and space frequency imbalance due to line interface circuit. The space match score is original score +

CORR_ADJ_S*	Match filter space frequency correlation adjust value polarity. 0: increase space correlation. 1: decrease space correlation.
CF_ADJ_SW	Carrier frequency detect count for software mode. When set the default carrier detect frequency values are over-ridden and CDB_FREQ_LOW_CNT (0x190-0x191) CDB_FREQ_HIGH_CNT (0x192-0x193) registers are used to determine CDB frequency detection. When reset default values are used.
SYM_ADJ_MD*	This bit is valid only when DEC_MD_SEL=1. 0: Symbol adjustment from first start bit. Only supports type 1 FSK signal. 1: Symbol adjustment applies to every start bit. Supports type 1 and type 2 FSK signal. (Default)
DEC_MD_SEL*	0: Data demodulation is triggered on START bit transition and local baud rate clock is used to decode symbol. 1: Data demodulation is triggered on START bit transition and micro adjustment is performed in symbol. (Default)

8.13.4 FSKD_STATUS

Address	Access Mode	Value At Reset	Nominal Value
0x183	R	0x02	

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	CDB	FSK_DF_OUT

FSK_DF_OUT	This bit is serial FSK data demodulated by hardware.
CDB	Carrier Detect status. 1: No carrier present. 0: Carrier detected.

8.13.5 FSKD_THRES

Address	Access Mode	Value At Reset	Nominal Value
0x184-0x185	R/W	0x0080	

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
84	FSKD_THRES[15:8]							
85	FSKD_THRES[7:0]							

FSKD_THRES[15:0] Sets the threshold of sensitivity for FSK signal detection. The peak level of input signal must be greater than threshold for valid detection. The threshold is given by FSKD_THRES/2^{19}*V_{pp}FS, where V_{pp}FS is the full

((-1)^(CORR_ADJ_S))*CORR_ADJ[2:0]. This will affect the decoded mark and space duration in the demodulated data. If space correlation matching score is increased, the 0 duration will be longer. Use of this mode is generally not necessary. It is used to compensate for system level imperfection if twist requirements can not be met with default values.

* It is recommended that SYM_ADJ_MD and DEC_MD_SEL remain at default values (1,1).

scale input peak-to-peak voltage at the ADC which is Avdd V. To refer to input voltage, adjust for the TI input gain.

8.13.6 FSKD_JUDGE

Address	Access Mode	Value At Reset	Nominal Value
0x0186	R/W	0x04	

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Reserved	Reserved	Reserved	Reserved	Reserved	FSK_CMP_CNT[2:0]		

FSK_CMP_CNT[2:0] Determines how many sample periods to examine to calculate the maximal match value.

Table 8-16 FSK_CMP_CNT Definition.

FSK_ CMP_CNT [2:0]	Polynomial
0x00	Y(n)
0x01	Max{ Y(n), Y(n-1) }
0x02	max{ Y(n),Y(n-1), …,Y(n-3) }
0x03	max{ Y(n),Y(n-1), ….,Y(n-4) }
0x04	max{ Y(n),Y(n-1),….,Y(n-5) }
0x05	max{ Y(n),Y(n-1),….,Y(n-6) }
0x06	max{ Y(n),Y(n-1),….,Y(n-7) }
0x07	max{ Y(n),Y(n-1),….,Y(n-9) }

8.13.7 FSKD_SYNC

Address	Access Mode	Value At Reset	Nominal Value
0x187	R/W	0x02	

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Reserved	Reserved	Reserved	Reserved	SYNC_NUM[3:0]			

SYNC_NUM[3:0] Determines how many synchronization patterns (0x55) should be detected before CDB (Carrier Detect Bar) goes low.

8.13.8 FSKD_CDET

Address	Access Mode	Value At Reset	Nominal Value
0x188	R/W	0x03	

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Reserved	Reserved	CDB_TIME[5:0]					

CDB_TIME[5:0] Determines the time before CDB (Carrier Detect Bar) goes low. Time is given by 8*CDB_TIME ms.

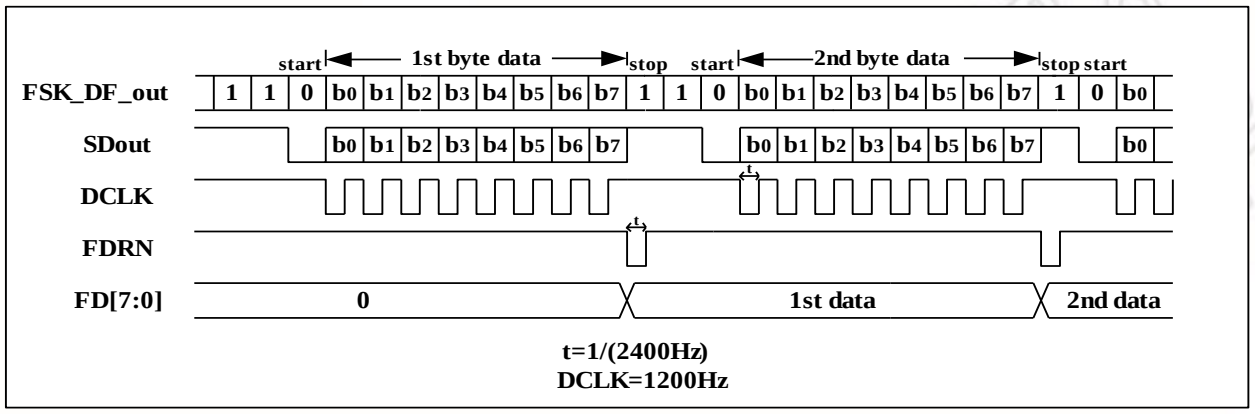
Carrier Detection

The carrier detection block determines whether there is an FSK carrier present on the input signal. The circuit examines an 8ms time slice and determines whether a carrier is present. CDB (Carrier Detect Bar) will go low under either of the following situations:

1. Carrier is detected for CDB_TIME*8ms.
2. SYNC_NUM synchronization bytes are detected.

Trecover

The Trecover block detects STOP-START bit transitions and generates bit and FIFO clocks to recover the transmitted data.



8.13.9 FSKD_FIFO_CTRL

Address	Access Mode	Value At Reset	Nominal Value
0x189	R/W	0x00	

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
RESERVED	FIFO_LEN			RESERVED	RESERVED	RESERVED	CLEAR

CLEAR Setting this bit resets the FSK receive FIFO.

FIFO_LEN Set FIFO interrupt length. Circuit will issue interrupt when (FIFO_LEN+1) samples are present in the FIFO.

8.13.10 FSKD_FIFO_DOUT

Address	Access Mode	Value At Reset	Nominal Value
0x18A	R	0x00	

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
FSKD_FIFO_DOUT							

FSKD_FIFO_DOUT	Received data stored in FIFO. The FIFO depth is 8 bytes. If FIFO is not read after a FIFO FULL condition then next data received will be lost. Reading FSKD_FIFO_DOUT resets the FSKD interrupt.

8.13.11 FSKD_FIFO_STATUS

Address	Access Mode	Value At Reset	Nominal Value
0x18B	R	0x01	

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
FIFO_DEPTH				PARITY_ERR	OVF	FULL	EMPTY

EMPTY	FIFO empty status. Set when FIFO is empty. Reset when FIFO is not empty.
FULL	FIFO full status. Set when FIFO is full.
OVF	FIFO overflow status bit. Set when FIFO is full and data has been lost. Reset when FIFO is empty.
PARITY_ERR	The parity error status bit belongs to the current FIFO data. If set then current head of FIFO has a parity error.
FIFO_DEPTH	FSK FIFO depth. This is the current number of samples in the FSK receive FIFO. Value lies in range 0 to 8.

8.13.12 FSKD_ENERGY_HIGH_TH

Address	Access Mode	Value At Reset	Nominal Value
0x18C	R/W	0x0C	

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
FSKD_ENERGY_HIGH_TH[7:0]							

FSKD_ENERGY_HIGH_TH[7:0]	Used for Bell 202, V.23 type1 standard. Sets the FSK carrier on energy high threshold. Detector will enter carrier on condition when FSK signal power is over this level for 3*FSKD_ENERGY_TC*1/16K seconds. For default FSKD_ENERGY_TC = 64 this is 12ms.

8.13.13 FSKD_ENERGY_LOW_TH

Address	Access Mode	Value At Reset	Nominal Value
0x18D	R/W	0x06	

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
FSKD_ENERGY_LOW_TH[7:0]							

FSKD_ENERGY_LOW_TH[7:0]	Used for Bell202, V.23 type1 standard. Sets the FSK carrier off energy low threshold. Detector will enter carrier off condition when FSK signal power is under this level for 2*FSKD_ENERGY_TC*1/16K seconds. For default FSKD_ENERGY_TC = 64 this is 8ms.

8.13.14 FSKD_ENERGY_TC

Address	Access Mode	Value At Reset	Nominal Value
0x18E	R/W	0x40	

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
FSKD_ENERGY_TC[7:0]							

FSKD_ENERGY_TC[7:0]	Used for bell202, v.23 type1 standard. Sets the FSK carrier energy accumulate period. The period = FSKD_ENERGY_TC * 1/16K seconds.

8.13.15

8.13.16 FSKD_CDB_FREQ_LOW_CNT

Address	Access Mode	Value At Reset	Nominal Value
0x190~0x191	R/W	0x2F14	

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
90	FSKD_CDB_FREQ_LOW_CNT[15:8]							
91	FSKD_CDB_FREQ_LOW_CNT[7:0]							

FSKD_CDB_FREQ_LOW_CNT	CDB low frequency detect count. Valid only when CF_ADJ_SW (0x182) is 1.

8.13.17 FSKD_CDB_FREQ_HIGH_CNT

Address	Access Mode	Value At Reset	Nominal Value
0x192~0x193	R/W	0x23F5	

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
92	FSKD_CDB_FREQ_HIGH_CNT[15:8]							
93	FSKD_CDB_FREQ_HIGH_CNT[7:0]							

FSKD_CDB_FREQ_HIGH_CNT	CDB high frequency detect count. Valid only when CF_ADJ_SW (0x182) is 1.

The formula for CDB_FREQ_CNT is:

$$CDB_FERQ_CNT = \frac{12288000}{FREQ(1+deviation\%)}$$

For example: Bell103 low band space frequency=1070Hz, deviation=-5%,

CDB_FREQ_CNT=12288000/(1070*(1-5%)) ≈ 12088(dec) = 2F38(hex)

Note: These two counts should be carefully set, or CDB detection will not work correctly. Normally, 5% deviation is suitable. If CDB is hard to detect, loosening these two count values will make CDB detection more sensitive but less accurate.

8.13.18 FSKD_LIMIT_TH

Address	Access Mode	Value At Reset	Nominal Value
0x194	R/W	0xFF	

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
FSKD_LIMIT_TH[7:0]							

FSKD_LIMIT_TH[7:0]*	Apply a clamp limiter to FSK signal. Input signal will be clamped to the region specified by this value. It is recommended to leave this at default value (no clamping).

* The clamp region is: { {FSKD_LIMIT_TH[7:0],10'b0} , (-1)* {FSKD_LIMIT_TH[7:0],10'b0} }. For example: FSKD_LIMIT_TH=0xFF, the clamp region is {18'h3FC00, (-1)*18'h3FC00 }

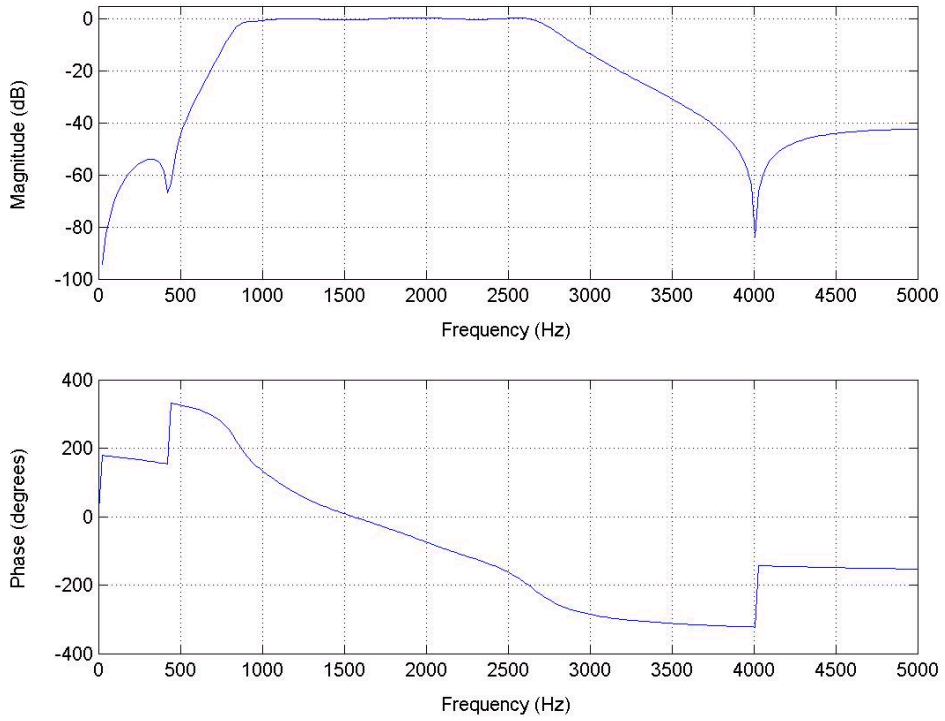


Figure 8-17 band pass filter frequency response

8.13.19 Example FSK Detector Usage

Usage of the FSK detector normally involves three steps:

1. Configure the detector.
2. Detect channel seizure.
3. Receive transmitted data.

```
// Configure Input path: Enable a TI input and set input gain.
writeCfgReg(CFG_TI_GAIN,0x0a);
writeCfgReg(CFG_TI_CTRL,0x82);
// Enable Line CODEC
writeCfgReg(CFG_LC_EN,0xa0);
// Enable FSK Detector
writeCfgReg(CFG_FSKD_CTRL,0x80);
// Set mode to Bell 103 1270/1070Hz
writeCfgReg(CFG_FSKD_MODE,0x04);
// Clear FIFO set interrupt level to 1 byte.
writeCfgReg(CFG_FSKD_FIFO_CTRL,0x01);
// Adjust thresholds if desired, for example:
writeCfgReg(CFG_FSKD_ENERGY_HI_TH,0x0C);
```

writeCfgReg(CFG_FSKD_ENERGY_TC,0x40);

Below is the Evaluation board interrupt service routine for reference. When triggered the state machine looks for a carrier synchronization signal then receives data to a FIFO. A 10ms timer tick is assumed in operation and correct path and specification configuration.

```
PUBLIC void vFSKDetFSM(void)
{
        static byte byCounter;
        byte tmp;

        switch(byFSKDetState)
        {
                case FSKD_IDLE:
                if(bUsbCtrl[0] & USB_CTRL0_FSK_DET_START)
                {
                        byFSKDetState = FSKD_CHANNEL_SEIZURE;
                }
                break;
                case FSKD_CHANNEL_SEIZURE:
                // The assumption here is that the FSK encoder block
                // general configuration has been setup already.
                // Here we configure for carrier detection.
                tmp = ReadRegister(CFG_FSKD_CTRL);
                tmp |= CFG_FSKD_CTRL__FSK_INT_MD | CFG_FSKD_CTRL__FSK_EN;
                writeCfgReg(CFG_FSKD_CTRL,tmp);
                // Clear FIFO and set interrupt to 6 bytes.
                writeCfgReg(CFG_FSKD_FIFO_CTRL,CFG_FSKD_FIFO_CTRL__CLEAR + 0x30);
                byCounter = 0;
                bUsbStatus[4] |= USB_STATUS4__FSK_WAIT;
                byFSKDetState = FSKD_CHANNEL_SEIZURE2;
                break;

                case FSKD_CHANNEL_SEIZURE2:
                if(DevStatus[2] & STATUS2_FSK_D)
                {
                        // We get here FIFO interrupt has occurred.
                        // We are in SYNC mode so just clear the FIFO
                        tmp = ReadRegister(CFG_FSKD_FIFO_DOUT);
                        writeCfgReg(CFG_FSKD_FIFO_CTRL,CFG_FSKD_FIFO_CTRL__CLEAR + 0x30);
                        byCounter = 0;
                        // Ensure that carrier detect has triggered
                        tmp = ReadRegister(CFG_FSKD_STATUS);
                        if(!(tmp & CFG_FSKD_STATUS__CDB))
                        {
                                bUsbStatus[4] |= USB_STATUS4__FSK_DETECTED;
                                bUsbStatus[4] &= ~USB_STATUS4__FSK_WAIT;
                                // FSK_INT_MD interrupt only occurs once.

                        }
```

```
                }
                else if((DevStatus[3] & STATUS3_TIMER_INT) && (bUsbStatus[4] &
USB_STATUS4__FSK_DETECTED))
                {
                        // Timer interrupt has occurred
                        tmp = ReadRegister(CFG_FSKD_FIFO_STATUS);
                        if(!(tmp & CFG_FSKD_FIFO_STATUS__EMPTY))
                        {
                                // If FIFO is not empty
                                tmp = ReadRegister(CFG_FSKD_FIFO_DOUT);
                                writeCfgReg(CFG_FSKD_FIFO_CTRL,CFG_FSKD_FIFO_CTRL__CLEAR + 0x30);
                                byCounter = 0;
                                break;
                        }
                        byCounter++;
                        // We wait 50ms without a FIFO interrupt then assume that we have
                        // finished synchronization
                        if(byCounter  > 4)
                        {
                                byCounter = 0;
                                // Switch to normal interrupt mode
                                tmp = ReadRegister(CFG_FSKD_CTRL);
                                tmp &= ~CFG_FSKD_CTRL__FSK_INT_MD;
                                writeCfgReg(CFG_FSKD_CTRL,tmp);
                                byFSKDetState = FSKD_DATA_RX;
                        }
                }
                break;
                case FSKD_DATA_RX:
                // Getting here means we are in the data acquisition phase of
                // transmission.
                if(DevStatus[2] & STATUS2_FSK_D)
                {
                        // We get here FIFO interrupt has occurred. There are at
                        // least 4 bytes on FIFO.
                        byCounter = 0;
                        tmp = ReadRegister(CFG_FSKD_FIFO_STATUS);
                        while(!(tmp & CFG_FSKD_FIFO_STATUS__EMPTY) &&  (RxDataFifo.Depth <
MAX_FIFO_SIZE))
                        {
                                if(tmp & (CFG_FSKD_FIFO_STATUS__OVF |
CFG_FSKD_FIFO_STATUS__PARITY_ERR))
                                {
                                        // Set a flag here to indicate transmission error.
                                        bUsbStatus[4] |= USB_STATUS4__RX_ERR;
                                }
                                tmp = ReadRegister(CFG_FSKD_FIFO_DOUT);
                                RxDataFifo.Fifo_Buffer[RxDataFifo.WriteIndex++ & 0x3F]= tmp;
                                RxDataFifo.Depth++;
                                tmp = ReadRegister(CFG_FSKD_FIFO_STATUS);
                        }
                }
```

```
            else if(DevStatus[3] & STATUS3_TIMER_INT)
            {
                    // We are here due to a timer tick. This means we may be finished with
                    // transmission.
                    byCounter++;
                    // We wait 50ms without a FIFO interrupt then we may have
                    // finished data transmission
                    if(byCounter  > 4)
                    {
                                    // First read off any data in FIFO
                            tmp = ReadRegister(CFG_FSKD_FIFO_STATUS);
                            while(!(tmp & CFG_FSKD_FIFO_STATUS__EMPTY) &&  (RxDataFifo.Depth <
MAX_FIFO_SIZE))
                            {
                                    if(tmp & (CFG_FSKD_FIFO_STATUS__OVF |
CFG_FSKD_FIFO_STATUS__PARITY_ERR))
                                    {
                                    // Set a flag here to indicate transmission error.
                                            bUsbStatus[4] |= USB_STATUS4__RX_ERR;
                                    }
                                    tmp = ReadRegister(CFG_FSKD_FIFO_DOUT);
                                    RxDataFifo.Fifo_Buffer[RxDataFifo.WriteIndex++ & 0x3F]= tmp;
                                    RxDataFifo.Depth++;
                                    tmp = ReadRegister(CFG_FSKD_FIFO_STATUS);
                            }
                            if(tmp & CFG_FSKD_FIFO_STATUS__EMPTY)
                            {
                            // We have an empty FIFO. If we have lost carrier then
                            // we can exit.
                                    tmp = ReadRegister(CFG_FSKD_STATUS);
                                    if(tmp & CFG_FSKD_STATUS__CDB)
                                    {
                                    // No carrier
                                            byFSKDetState = FSKD_WAIT_IDLE;
                                    // Report back that Carrier lost
                                            bUsbStatus[4] &= ~USB_STATUS4__FSK_DETECTED;
                                    }
                            }
                    }
            }
            break;
            case FSKD_WAIT_IDLE:
            if(!(bUsbCtrl[0] & USB_CTRL0_FSK_DET_START))
            {
                    byFSKDetState = FSKD_IDLE;
                    bUsbStatus[4] &= ~(USB_STATUS4__FSK_WAIT | USB_STATUS4__RX_ERR |
USB_STATUS4__FSK_DETECTED);
            }

            break;
            default:
            byFSKDetState = FSKD_IDLE;
```

```
                break;
        }

}
```

```
// Setup channel seizure detection
OrCfgReg(CFG_FSKD_CTRL,0x01);
AndCfgReg(CFG_FSKD_MODE,~0x20);

// Wait for INT for carrier detection.
// Read off data from FIFO of channel seizure or wait until channel seizure time has
// elapsed then reset FIFO. Channel now transmits STOP (MARK) signal for some period

// Change mode to appropriate packaged mode.
AndCfgReg(CFG_FSKD_CTRL,~0x01);
OrCfgReg(CFG_FSKD_MODE,0x20);    // parity enable

// Service FIFO interrupt until desired data collected or CDB goes high indicating end
// of transmission.
```

8.14 CAS and Arbitrary Tone (ATD) Detector

The CAS and Arbitrary tone detector (ATD) allows detection of CAS tones and is configurable to detect arbitrary tones. CAS (CPE Alert Signal) is used to signal the CPE (Customer Premise Equipment) while off hook that the exchange wishes to send some alert signal the end user, such as call waiting. It is used only in off-hook signaling in the US Bellcore system and ETSI system, but in the UK BT system it is used for both on and off-hook signaling. Figure 8-18 shows the architecture of the CAS detector.

新唐科技 NUVOTON INTELLECTUAL PROPERTY

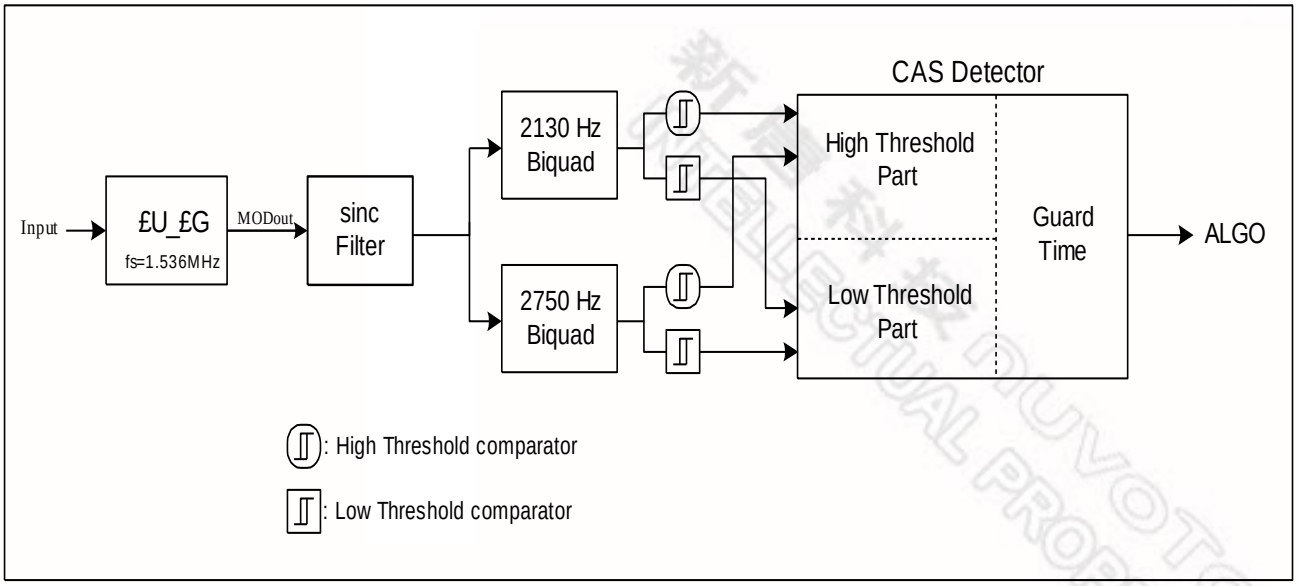


Figure 8-18 Block Diagram of CAS Detection

The CAS detector consists of two biquad bandpass filters to extract the high and low tones of the CAS signal. The characteristics of these two filters are shown Figure 8-19 and Figure 8-20. Their frequency response is:

- High tone (2750Hz) band pass filter: 2600Hz ≤ f ≤ 3000Hz
- Low tone (2130Hz) band pass filter: 2020Hz ≤ f ≤ 2200Hz

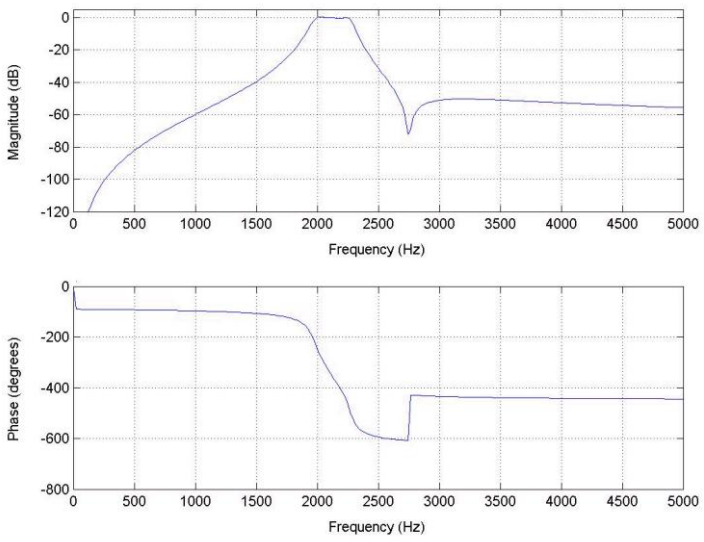


Figure 8-19 Detector Biquad Low Tone Frequency Response

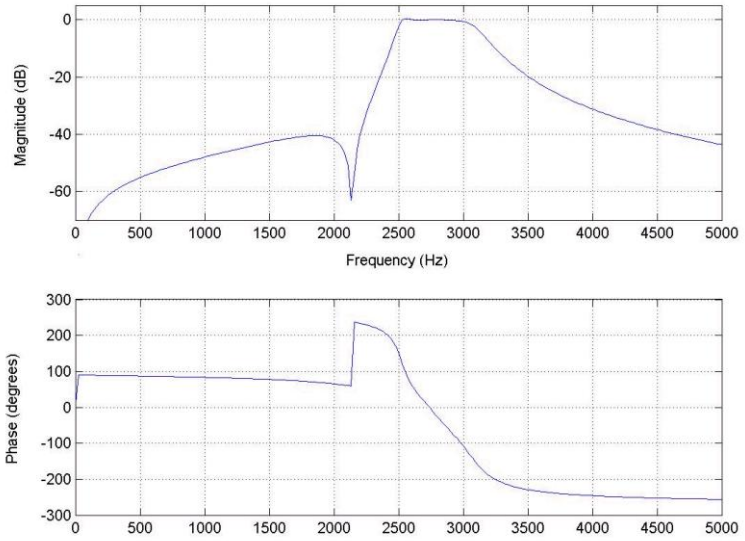


Figure 8-20 CAS Detector Biquad High Tone Frequency Response

The CAS detector can also be programmed as an arbitrary tone detector (ATD). In ATD mode coefficients for the low and high tone biquad filters are provided by the user. The frequency specifications and deviation are also programmed by the user. The ATD can be configured to detect single tonesor dual tones.

8.14.1 CAS_CTRL

Address	Access Mode	Value At Reset	Nominal Value
0x200	R/W	0x40	

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CAS_EN	IIR_EN	TONE_ON_INT	TONE_OFF_INT	RESERVED	allow_noise	FREQ_DEV	

CAS_EN	1: CAS Detector enabled. 0: CAS detector powered down.
IIR_EN	Enable internal IIR filter. IIR_EN must be cleared before accessing filter coefficients in arbitrary tone detection (ATD) mode. The response of the IIR filter in CAS mode is shown Figure 8-21 CAS Internal IIR Filter Frequency Response Figure . The response of the filter in ATD mode is determined by the user coefficient loaded into RAM. Once filter coefficients are loaded into RAM IIR_EN must be set active once more.
TONE_ON_INT	1: Enable 0: disable interrupt generation for tone present detection.
TONE_OFF_INT	1: Enable 0: disable interrupt generation for tone absent detection.
allow_noise	1: Tolerating noise signal existence. 0: Not tolerating noise signal existence.
FREQ_DEV[1:0]	Controls the acceptable frequency deviation of CAS detection as per Table 8-17. Not applicable in ATD mode.

Table 8-17 CAS Detector Frequency Deviation Control.

FREQ_DEV[1:0]	00	01	10	11
% Dev	2%	1%	2%	3%
High Tone 2750 Hz	2805~2695 Hz	2777~2722 Hz	2805~2695 Hz	2832~2667 Hz
Low Tone 2130 Hz	2176~2087 Hz	2151~2109 Hz	2176~2087 Hz	2193~2066 Hz

8.14.2 CAS_THRES_LOW – CAS detector low threshold

Address	Access Mode	Value At Reset	Nominal Value
0x201-0x202	R/W	0x0000	

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
01	CAS_THRES_LOW[15:8]							
02	CAS_THRES_LOW[7:0]							

CAS_THRES_LOW[15:0] Determines the low boundary hysteresis of the CAS/ATD signal detection. The value is in 2's complement 1.15 format. The value 0x7FFF corresponds to full swing input ($1V_{pp}$ at ADC input). For example:

CAS_THRES_LOW = 0x020C => 0000 0010 0000 1100b

$2^{(-7)} + 2^{(-13)} + 2^{(-14)} \approx 8mV_{pp}$

8.14.3 CAS_MULT

Address	Access Mode	Value At Reset	Nominal Value
0x203	R/W	0x11	

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Reserved	cas_src	det_rsol		MULT[3:0]			

cas_src Select the source of cas input. 0: select from sinc filter. 1: select from digital filter.

det_rsol The resolution of detecting CAS signal

00: 48KHz sampling clock generated by internal counter

01: 48KHz sampling clock generated by internal counter which aligned with detected CAS signal .

10: 64KHz sampling clock generated by internal counter which aligned with detected CAS signal.

11: 96KHz sampling clock generated by internal counter which aligned with detected

CAS signal.

MULT [3:0] The CAS/ATD detector processes two thresholds, CAS_THRES_LOW and CAS_THRES_HIGH. CAS_THRES_HIGH is calculated by CAS_THRES_LOW << MULT. For example above and MULT=4

CAS_THRES_HIGH = 8 mV_{pp} * 2^4 =128 mV_{pp}

If (CAS_THRES_LOW << MULT) ≥ 0x7FFF, CAS_THRES_HIGH saturates to 0x7FFF.

For a valid detection the peak signal level must fall between CAS_THRES_LOW and CAS_THRES_HIGH.

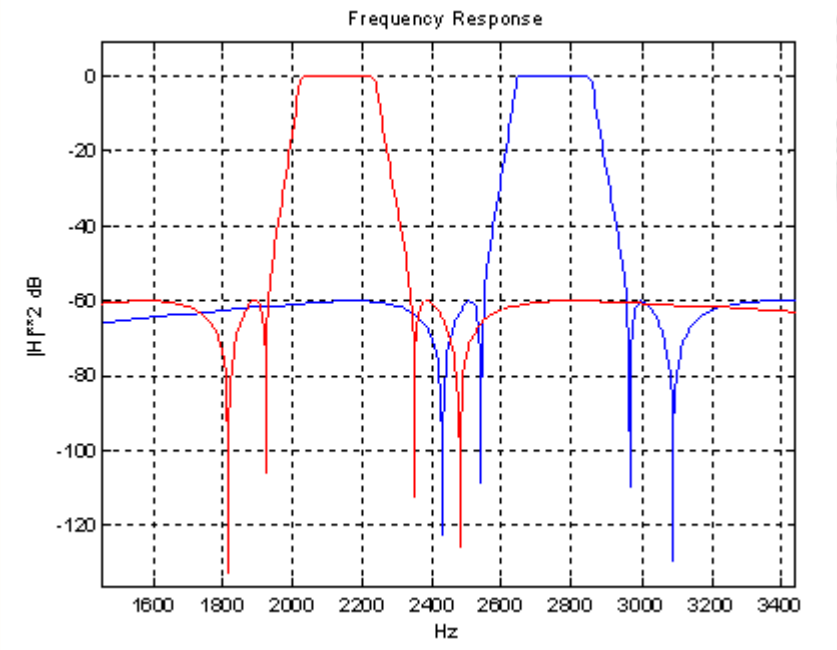


Figure 8-21 CAS Internal IIR Filter Frequency Response Figure

8.14.4 CAS_PRESENT

Address	Access Mode	Value At Reset	Nominal Value
0x204	R/W	0x40	

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CAS_PRES_TIME							

CAS_PRES_TIME [7:0] This is the time a detected CAS/ATD tone must be present to generate a CAS interrupt. The time equals CAS_PRES_TIME * 0.5ms.

8.14.5 CAS_ABSENT

Address	Access Mode	Value At Reset	Nominal Value
0x205	R/W	0x20	

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CAS_ABSENT_TIME							

CAS_ABSENT_TIME [7:0] This is the time a detected CAS/ATD tone must be absent before removal of CAS_DET signal. The time equals CAS_ABSENT_TIME * 0.5ms. This parameter is used to mask short drop outs of the CAS signal.

8.14.6 CAS_STATUS

Address	Access Mode	Value At Reset	Nominal Value
0x206	R	0x00	

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	CAS_ADT_DET	

CAS_ADT_DET Indicates what type of tone is detected as per Table 8-18.

Table 8-18 CAS/ADT Detection Bits

CAS_MODE[6:5] (DET_MODE)	CAS_ADT_DET[1]	CAS_ADT_DET[0]
00	Don't Care	Dual Tone Status
01	Don't Care	Low Tone Status
10	Don't Care	High Tone Status
11	High Tone Status	Low Tone Status

8.14.7 CAS_MODE - Arbitrary Tone Detection Mode

Address	Access Mode	Value At Reset	Nominal Value
0x0207	R/W	0x10	

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ATD_MODE	DET_MODE		STATS				

ATD_MODE 0: CAS detection mode. 1: Arbitrary Tone Detection (ATD) mode.
In ATD_MODE the detection and filter parameters are programmed by software to detect arbitrary tone.

DET_MODE

00: dual-tone mode
01: detect low tone only when ATD_MODE=1
10: detect high tone only when ATD_MODE=1
11: detect high/low tone separately when ATD_MODE=1

STATS

Number of sine wave periods to be used for frequency calculations. The more cycles the higher the frequency accuracy but the more computation time required. This value must be set to 0x01~0x1E range.

For ATD mode, user can input the value from 1 to 1E.

For CAS mode, the STATS value is decided by det_rsol(0x203[5:4]).

det_rsol	STATS
00	11(h)
01	11(h)
10	d(h)
11	9(h)

8.14.8 ATD_MAX_HFC (ATD_MODE = 1)

Address	Access Mode	Value At Reset	Nominal Value
0x0208-0x209	R/W	0x0110	

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	ATD_MAX_HFC[9:8]	
ATD_MAX_HFC[7:0]							

ATD_MAX_HFC

This is the maximum frequency count for the high tone arbitrary frequency detector. A tone is detected when the frequency counter falls between ATD_MIN_HFC and ATD_MAX_HCF. The value of this register should be set to:

ATD_MAX_HFC = (SMPL_CLK * STATS) / (Desired_High_Frequency * (1+ error%)),

SMPL_CLK is the frequency of sampling clock defined by register det_rsol register.

8.14.9 ATD_MIN_HFC (ATD_MODE = 1)

Address	Access Mode	Value At Reset	Nominal Value
0x020A-0x20B	R/W	0x0120	

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	ATD_MIN_HFC[9:8]	
ATD_MIN_HFC[7:0]							

ATD_MIN_HFC

This is the minimum frequency count for the high tone arbitrary frequency detector. A tone is detected when the frequency counter falls between ATD_MIN_HFC and ATD_MAX_HCF. The value of this register should be set to:

ATD_MIN_HFC = SMPL_CLK * STATS) / (Desired_High_Frequency * (1- error%)), SMPL_CLK is

the frequency of sampling clock defined by register det_rsol.

8.14.10 ATD_MAX_LFC (ATD_MODE = 1)

Address	Access Mode	Value At Reset	Nominal Value
0x020C-0x20D	R/W	0x0161	

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	ATD_MAX_LFC[9:8]	
ATD_MAX_LFC[7:0]							

ATD_MAX_LFC

This is the maximum frequency count for the low tone arbitrary frequency detector. A tone is detected when the frequency counter falls between ATD_MIN_LFC and ATD_MAX_LCF. The value of this register should be set to:

ATD_MAX_LFC = (SMPL_CLK * STATS) / (Desired_low_tone_Frequency * (1+ error%)), SMPL_CLK is the frequency of sampling clock defined by register det_rsol.

8.14.11 ATD_MIN_LFC (ATD_MODE = 1)

Address	Access Mode	Value At Reset	Nominal Value
0x020E-0x20F	R/W	0x016F	

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	ATD_MIN_LFC[9:8]	
ATD_MIN_LFC[7:0]							

ATD_MIN_LFC

This is the minimum frequency count for the high tone arbitrary frequency detector. A tone is detected when the frequency counter falls between ATD_MIN_HFC and ATD_MAX_HCF. The value of this register should be set to:

ATD_MIN_LFC = (SMPL_CLK * STATS) / (Desired_low_tone_Frequency * (1- error%)), SMPL_CLK is the frequency of sampling clock defined by register det_rsol.

8.14.12 External Dual-IIR Coefficient (ATD_MODE= 1)

Address	Access Mode	Value At Reset	Nominal Value
0x0210~0241	R/W	0x00	

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ATD_HF_COEFF							

Address	Access Mode	Value At Reset	Nominal Value
0x0242~0273	R/W	0x00	

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ATD_LF_COEFF							

ATD_HF_COEFF IIR filter coefficients the high tone arbitrary tone detector.

ATD_LF_COEFF IIR filter coefficients the low tone arbitrary tone detector.

Before accessing these coefficients (both for read or write), 0x0200[6] IIR_EN must be cleared and 0x0200[7] CAS_EN must be set. This allows the internal IIR filter coefficients to be replaced by ones of the user's choice for arbitrary tone detection. Once loaded IIR_EN is set active once more for operation.

The input to the CAS detector consists of two cascaded IIR filters, each of them are **5-order biquad** filters (10-order IIR filter) and operate at a **16kHz** sampling **rate**.

The user must set each biquad filter coefficient according to a specific **order b0 b1 a1 b2 a2 with** respect to the following formula:

y(n) = b0*x(n) + b1*x(n-1) + b2*x(n-2) – a1*y(n-1) – a2*y(n-2)

Each parameter is **16 bits** long in a **3.13 format**, and is stored MSB first. Thus for high tone filter, b0[15:8] is at address 0x210, b0[7:0] is at 0x211, b1[15:8] is at 0x212 etc..

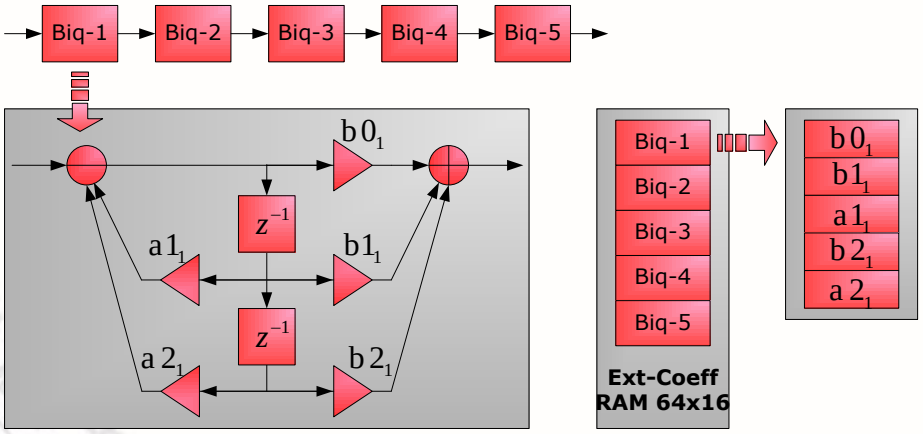


Figure 8-22 FSK ATD IIR filter structure

8.14.13 FSK_COEFF – FSK Encoder Coefficient RAM Data

Address	Access Mode	Value At Reset	Nominal Value
0x0274	R/W	0x00	

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
FSK_COEFF							

The FSK encoder has a built-in 8th-order IIR filter formed by cascading 4 biquad filters. There are total up to 40 bytes to be programmed and the format is the same as CAS/ATD filter.

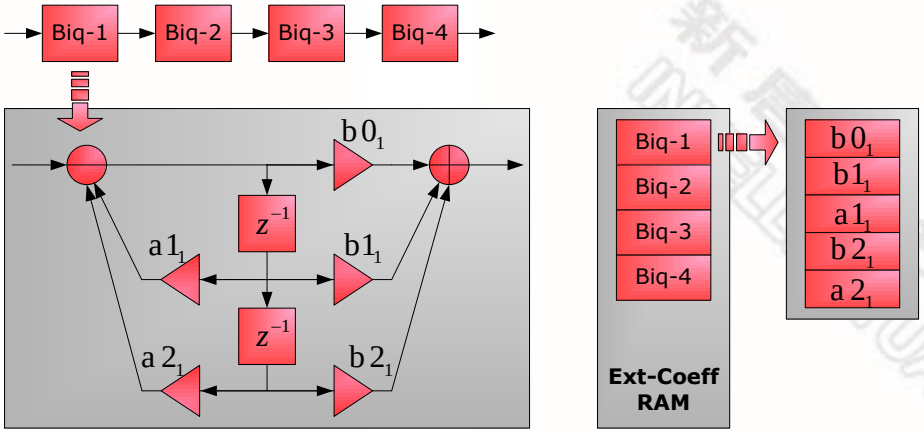


Figure 8-23 FSK encoder IIR filter structure

In order to program this ram, 0x0200[7] CAS_EN must be set, and 0x0200[6] IIR_EN must be clear

The parameter is 16 bits long with 3.13 format, and user must write high byte first then low byte.

0x0275[0]=0 is the high byte, and 0x0275[0] =1 is the low byte. The RAM for these filter coefficients is accessed through register FSK_COEFF and addressed through FSK_COEFF_ADDR

When software reads or writes register FSK_COEFF (0x274), the FSK_COEFF_ADDR (0x275) register will automatically increment. When the address reaches 0x27 it will reset to 0x00.

User can program _COEFF_ADDR for choosing specific FSK encoder RAM content both for read and write.

Care must be taken that only 0x0275[0] = 1, the 16-bit coefficient data will be written into ram.

8.14.14 FSK_COEFF_ADDR - FSK Encoder Coefficient RAM Address

Address	Access Mode	Value At Reset	Nominal Value
0x0275	R/W	0x00	

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
FSK_COEFF_ADDR							

By setting this address, software can access up to 40 bytes via 0x0274.

8.14.15 CAS_MAX_HFC (FINE_TUNE = 1)

Address	Access Mode	Value At Reset	Nominal Value
0x276-0x277	R/W	0x010E	

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	CAS_MAX_HFC[9:8]	
CAS_MAX_HFC[7:0]							

CAS_MAX_HFC	This is the maximum frequency count for the high tone arbitrary frequency detector. A tone is detected when the frequency counter falls between CAS_MIN_HFC and CAS_MAX_HCF. The value of this register should be set to: CAS_MAX_HFC = (48000 * 16) / (Desired_High_Frequency * (1+ error%))

8.14.16 CAS_MIN_HFC (FINE_TUNE = 1)

Address	Access Mode	Value At Reset	Nominal Value
0x278-0x279	R/W	0x0121	

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	CAS_MIN_HFC[9:8]	
CAS_MIN_HFC[7:0]							

CAS_MIN_HFC	This is the minimum frequency count for the high tone arbitrary frequency detector. A tone is detected when the frequency counter falls between CAS_MIN_HFC and CAS_MAX_HCF. The value of this register should be set to: CAS_MIN_HFC = (48000 * 16) / (Desired_High_Frequency * (1- error%))

8.14.17 CAS_MAX_LFC (FINE_TUNE = 1)

Address	Access Mode	Value At Reset	Nominal Value
0x27A-0x27B	R/W	0x015D	

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	CAS_MAX_LFC[9:8]	
CAS_MAX_LFC[7:0]							

CAS_MAX_LFC	This is the maximum frequency count for the low tone arbitrary frequency detector. A tone is detected when the frequency counter falls between CAS_MIN_LFC and CAS_MAX_LCF. The value of this register should be set to: CAS_MAX_LFC = (48000 * 16) / (Desired_low_tone_Frequency * (1+ error%))

8.14.18 CAS_MIN_LFC (FINE_TUNE = 1)

Address	Access Mode	Value At Reset	Nominal Value
0x27C-0x27D	R/W	0x0175	

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	CAS_MIN_LFC[9:8]	
CAS_MIN_LFC[7:0]							

CAS_MIN_LFC	This is the minimum frequency count for the high tone arbitrary frequency detector. A tone is detected when the frequency counter falls between CAS_MIN_HFC and CAS_MAX_HCF. The value of this register should be set to: CAS_MIN_LFC = (48000 * 16) / (Desired_low_tone_Frequency * (1- error%))

8.14.19

8.14.20

8.14.21 Arbitrary Tone Detector Example

Below is an example to setup the arbitrary tone detector to detect 1400 and 2300Hz tones.

```
// Enable ARB, disable IIR
writeCfgReg(CFG_CAS_CTRL,0xa1);
writeCfgReg(CFG_CAS_THRES_LOW,0x01);
writeCfgReg(CFG_CAS_MULT,0x02);
// 80ms present time
writeCfgReg(CFG_CAS_PRESENT,0xa0);
// 60ms absent time
writeCfgReg(CFG_CAS_ABSENT,0x78);
// ARB mode and both high and low detection
writeCfgReg(CFG_CAS_MODE,0xf0);
// 2300Hz +/- 3%
writeCfgReg(CFG_ATD_MAX_HFC,0x01);
writeCfgReg(CFG_ATD_MAX_HFC_LSB,0x44);
writeCfgReg(CFG_ATD_MIN_HFC,0x01);
writeCfgReg(CFG_ATD_MIN_HFC_LSB,0x58);
// 1400Hz +/- 3%
writeCfgReg(CFG_ATD_MAX_LFC,0x02);
```

```
writeCfgReg(CFG_ATD_MAX_LFC_LSB,0x14);
writeCfgReg(CFG_ATD_MIN_LFC,0x02);
writeCfgReg(CFG_ATD_MIN_LFC_LSB,0x36);
// Biquad coefficients for 2300 and 1400Hz
writeCfgReg(0x211,0x03); writeCfgReg(0x214,0xd8); writeCfgReg(0x215,0xd8);
writeCfgReg(0x216,0xff); writeCfgReg(0x217,0xfc); writeCfgReg(0x218,0x1f);
writeCfgReg(0x219,0x39); writeCfgReg(0x21a,0x20); writeCfgReg(0x21c,0xda);
writeCfgReg(0x21d,0xf6); writeCfgReg(0x21e,0xd9); writeCfgReg(0x21f,0x61);
writeCfgReg(0x220,0x20); writeCfgReg(0x222,0x1f); writeCfgReg(0x223,0x75);
writeCfgReg(0x224,0x20); writeCfgReg(0x226,0xd5); writeCfgReg(0x227,0xf7);
writeCfgReg(0x228,0xd8); writeCfgReg(0x229,0x08); writeCfgReg(0x22a,0x1f);
writeCfgReg(0x22b,0xff); writeCfgReg(0x22c,0x1f); writeCfgReg(0x22d,0x78);
writeCfgReg(0x22e,0x20); writeCfgReg(0x230,0xda); writeCfgReg(0x231,0x08);
writeCfgReg(0x232,0xd9); writeCfgReg(0x233,0x71); writeCfgReg(0x234,0x1f);
writeCfgReg(0x235,0xff); writeCfgReg(0x236,0x1f); writeCfgReg(0x237,0xd5);
writeCfgReg(0x238,0x20); writeCfgReg(0x23a,0xd6); writeCfgReg(0x23b,0xc8);
writeCfgReg(0x23c,0xd7); writeCfgReg(0x23d,0x85); writeCfgReg(0x23e,0x20);
writeCfgReg(0x240,0x1f); writeCfgReg(0x241,0xd6); writeCfgReg(0x243,0x03);
writeCfgReg(0x246,0xca); writeCfgReg(0x247,0x14); writeCfgReg(0x248,0xff);
writeCfgReg(0x249,0xfc); writeCfgReg(0x24a,0x1f); writeCfgReg(0x24b,0x3a);
writeCfgReg(0x24c,0x20); writeCfgReg(0x24e,0xcb); writeCfgReg(0x24f,0x36);
writeCfgReg(0x250,0xca); writeCfgReg(0x251,0x56); writeCfgReg(0x252,0x20);
writeCfgReg(0x254,0x1f); writeCfgReg(0x255,0x74); writeCfgReg(0x256,0x20);
writeCfgReg(0x258,0xc7); writeCfgReg(0x259,0xe5); writeCfgReg(0x25a,0xc9);
writeCfgReg(0x25b,0x6e); writeCfgReg(0x25c,0x20); writeCfgReg(0x25e,0x1f);
writeCfgReg(0x25f,0x7a); writeCfgReg(0x260,0x20); writeCfgReg(0x262,0xca);
writeCfgReg(0x263,0x8d); writeCfgReg(0x264,0xca); writeCfgReg(0x265,0x38);
writeCfgReg(0x266,0x20); writeCfgReg(0x268,0x1f); writeCfgReg(0x269,0xd5);
writeCfgReg(0x26a,0x20); writeCfgReg(0x26c,0xc8); writeCfgReg(0x26d,0x66);
writeCfgReg(0x26e,0xc8); writeCfgReg(0x26f,0xf0); writeCfgReg(0x270,0x1f);
writeCfgReg(0x271,0xff); writeCfgReg(0x272,0x1f); writeCfgReg(0x273,0xd7);
// Enable ARB, enable IIR
writeCfgReg(CFG_CAS_CTRL,0xe1);
```

An example interrupt service routine to use the arbitrary tone detector and DTMF and tone generator to perform a SIA contact ID transmission is given below. It assumes that ARB configuration is done as above.

```
PUBLIC void vContactIDFSM(void)
{
        static byte byCounter;
        static XDATA byte retry;
        byte tmp, i;

        switch(byContactIDState)
        {
                case CONTID_IDLE:
                if(bUsbCtrl[0] & USB_CTRL0_CONTID_START){
                        byCounter = 0;
                        retry =0;
                        byContactIDState = CONTID_HANDSHAKE0;
                        // Set timer to 1 second
                        writeCfgReg(CFG_TIME_CTRL,0x00);
                        writeCfgReg(CFG_TIME_TARG,0x03);
                        writeCfgReg(CFG_TIME_TARG_LSB,0xE7);
```

```
        writeCfgReg(CFG_TIME_CTRL,0x80);

}
break;
case CONTID_HANDSHAKE0:
// This is first stage of detecting ContactID handshake a
// 1400Hz tone.
if(DevStatus[2] & STATUS2_CAS_INT){
        tmp = ReadRegister(CFG_CAS_STATUS);
        if(tmp == CFG_CAS_STATUS__LOW_TONE){
                // Change timer to 10msec
                writeCfgReg(CFG_TIME_CTRL,0x00);
                writeCfgReg(CFG_TIME_TARG,0x00);
                writeCfgReg(CFG_TIME_TARG_LSB,0x09);
                writeCfgReg(CFG_TIME_CTRL,0x80);
                byCounter = 0;
                byContactIDState = CONTID_HANDSHAKE1;
        }
}
else if(DevStatus[3] & STATUS3_TIMER_INT){
        // Timer interrupt has occurred
        byCounter++;
        if(byCounter > 10){
                // 10 second timeout waiting for handshake tone
                byContactIDState = CONTID_WAIT_IDLE;
                // Change timer to default 10msec
                writeCfgReg(CFG_TIME_CTRL,0x00);
                writeCfgReg(CFG_TIME_TARG,0x00);
                writeCfgReg(CFG_TIME_TARG_LSB,0x09);
                writeCfgReg(CFG_TIME_CTRL,0x80);
        }
}
break;
case CONTID_HANDSHAKE1:
// This is second stage of detecting ContactID handshake a
// 2300Hz tone.
if(DevStatus[2] & STATUS2_CAS_INT){
        tmp = ReadRegister(CFG_CAS_STATUS);
        if(tmp == CFG_CAS_STATUS__HI_TONE){
                byCounter = 0;
                byContactIDState = CONTID_DELAY_TO_TX;
        }
}
else if(DevStatus[3] & STATUS3_TIMER_INT){
        // Timer interrupt has occurred
        byCounter++;
        if(byCounter > 100){
                // 100 millisecond timeout waiting for handshake tone
                bUsbStatus[4] &= ~USB_STATUS4__CTID_STATUS;
                bUsbStatus[4] |= USB_STATUS4__CTID_TIMEOUT1;
                byContactIDState = CONTID_WAIT_IDLE;
```

```
        }
}
break;

case CONTID_DELAY_TO_TX:
if(DevStatus[3] & STATUS3_TIMER_INT){
        // Timer interrupt has occurred
        byCounter++;
        if(byCounter > 199){
                // 200 millisecond delay until transmit
                byContactIDState = CONTID_TRANSMIT0;
        }
}
break;

case CONTID_TRANSMIT0:
// If there is Contact ID data on FIFO to
// transmit, start transmission.
// The 16 digits of the CID frame are packed into
// 8 bytes of the FIFO.
if(TxDataFifo.Depth >= 8){
        // Load DTMF tone buffer
        for(i=0;i<8;i++){
                writeCfgReg(CFG_TONE_BUFFER+i,
                TxDataFifo.Fifo_Buffer[((TxDataFifo.ReadIndex + i) & 0x3F)]);
        }
        writeCfgReg( CFG_TONE_BUF_LEN, 16);
        writeCfgReg( CFG_TONE_CTRL, CFG_TONE_CTRL__CLK_EN +
        CFG_TONE_CTRL__DTMF_MODE + CFG_TONE_CTRL__GEN_EN);
        byContactIDState = CONTID_TRANSMIT1;
}else{
        // No data so finished transmission
        bUsbStatus[4] &= ~USB_STATUS4__CTID_STATUS;
        bUsbStatus[4] |= USB_STATUS4__CTID_FINISHED;
        byContactIDState = CONTID_WAIT_IDLE;
}
break;
case CONTID_TRANSMIT1:
if(DevStatus[2] & STATUS2_TONE_INT){
        // DTMF transmission finished.
        byCounter=0;
        // Set timer to 250ms
        writeCfgReg(CFG_TIME_CTRL,0x00);
        writeCfgReg(CFG_TIME_TARG,0x00);
        writeCfgReg(CFG_TIME_TARG_LSB,0xF9);
        writeCfgReg(CFG_TIME_CTRL,0x80);
        byContactIDState = CONTID_KISSOFF0;
}
break;

case CONTID_KISSOFF0:
```

```
// Detecting 1400Hz Kiss-off tone.
if(DevStatus[2] & STATUS2_CAS_INT){
        tmp = ReadRegister(CFG_CAS_STATUS);
        if(tmp == CFG_CAS_STATUS__LOW_TONE){
                byCounter = 0;
                // Setup to generate tone OFF interrupt
                writeCfgReg(CFG_CAS_CTRL, CFG_CAS_CTRL__TONE_OFF_INT +
                CFG_CAS_CTRL__IRR_EN + CFG_CAS_CTRL__CAS_EN);
                byContactIDState = CONTID_KISSOFF1;
        }
}
else if(DevStatus[3] & STATUS3_TIMER_INT)
{
        // Timer interrupt has occurred
        byCounter++;
        if(byCounter > 5){
                // 1.25 sec timeout waiting for handshake tone
                writeCfgReg(CFG_TIME_CTRL,0x00);
                writeCfgReg(CFG_TIME_TARG,0x00);
                writeCfgReg(CFG_TIME_TARG_LSB,0x09);
                writeCfgReg(CFG_TIME_CTRL,0x80);
                if(retry < 4){
                        retry++;
                        byContactIDState = CONTID_TRANSMIT0;
                }else{
                        bUsbStatus[4] &= ~USB_STATUS4__CTID_STATUS;
                        bUsbStatus[4] |= USB_STATUS4__CTID_TIMEOUT2;
                        byContactIDState = CONTID_WAIT_IDLE;
                }
        }
}
break;

case CONTID_KISSOFF1:
// Detecting end of 1400Hz Kiss-off tone.
if(DevStatus[2] & STATUS2_CAS_INT)
{
        // Interrupt set to detect end of kiss-off tone.
        // Setup to generate tone OFF interrupt
        writeCfgReg(CFG_CAS_CTRL, CFG_CAS_CTRL__TONE_ON_INT +
        CFG_CAS_CTRL__IRR_EN + CFG_CAS_CTRL__CAS_EN);
                // We have successfully sent the Contact ID message
                // clear data from the FIFO and go to next message.
        TxDataFifo.ReadIndex += 8;
        TxDataFifo.Depth -= 8;
        retry = 0;
                // Change timer tick to 20ms to create 400ms delay

        writeCfgReg(CFG_TIME_CTRL,0x00);
        writeCfgReg(CFG_TIME_TARG,0x00);
        writeCfgReg(CFG_TIME_TARG_LSB,19);
```

```
                writeCfgReg(CFG_TIME_CTRL,0x80);
                byCounter = 0;
                byContactIDState = CONTID_DELAY_TO_TX;
            }
            break;

            case CONTID_WAIT_IDLE:
            if(!(bUsbCtrl[0] & USB_CTRL0_CONTID_START)){
                byContactIDState = CONTID_IDLE;
            }
            break;

            default:
            byContactIDState = CONTID_IDLE;
            break;
        }
}
```

8.15 Voice Energy Detection (Speech Energy Detection)

The voice energy detector allows measurement of signal energy in the voice band between 750Hz to 1400Hz. It is used to detect speech on the line to aid in CID functions or line in use detection. Band-pass filter frequency response of voice energy detection is in Figure 8-24.

.

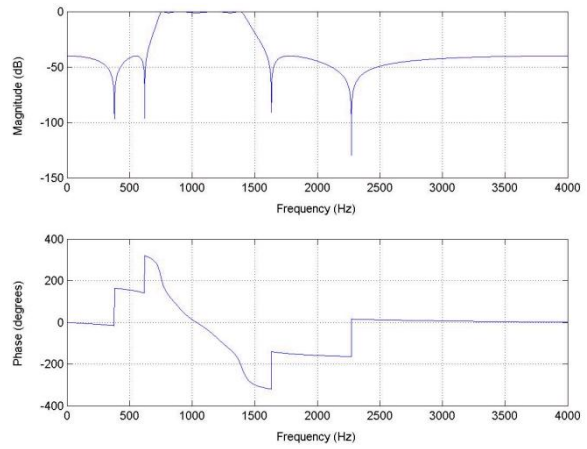


Figure 8-24 Band Pass filter frequency response of voice energy detection

8.15.1 VD_CTRL

Address	Access Mode	Value At Reset	Nominal Value
0x1F0	R/W	0x00	

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
VD_EN	Reserved	Reserved	Reserved	VD_TC [3:0]			

VD_TC[3:0] Voice Detect TC (time constant) determines the response of the voice detect energy averaging. Larger values imply a larger time constant, greater stability but less responsive to changes in energy level.

VD_EN Enable the Voice Detection block

8.15.2 VD_STATUS

Address	Access Mode	Value At Reset	Nominal Value
0x1F1	R	0x00	

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	VD_RX

VD_RX Voice Energy Detected. 1: Energy exceeds VD_THRES 0: Energy less than VD_THRES or VD_EN=0.

8.15.3 VD_THRES

Address	Access Mode	Value At Reset	Nominal Value
0x1F2~0x1F3	R/W	0x0100	

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
F2	VD_THRES [15:8]							
F3	VD_THRES [7:0]							

VD_THRES[15:0] This register defines the threshold of voice detection (speech energy). If the energy is greater than VD_THRES, VD_RX is high. The 16 bit value represents the decimal fraction of full scale in 2's complement format.

8.15.4 VD_ENERGY

Address	Access Mode	Value At Reset	Nominal Value
0x1F4~0x1F5	R	0x0000	

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
F4	VD_ENERGY [15:8]							
F5	VD_ENERGY [7:0]							

VD_ENERGY[15:0] This register represents the instantaneous signal power from 750Hz to 1400Hz. The value is in the range, 0x0000 to 0xFFFF, with larger value representing larger signal power.

8.16 Call Progress Tone Detector

The call progress tone detector is a narrow band pass filter from 300Hz to 650Hz; its detection algorithm is similar to voice energy detection. It can be used to detect call progress tones such as dial tones.

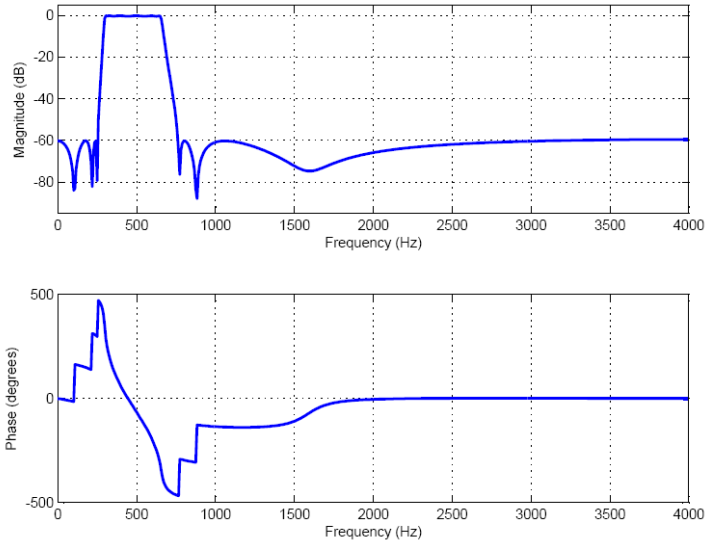


Figure 8-25 Call progress tone band-pass filter frequency response

8.16.1 CPT_CTRL

Address	Access Mode	Value At Reset	Nominal Value
0x280	R/W	0x07	

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
CPT_EN	Reserved	Reserved	Reserved	CPT_TC [3:0]			

CPT_TC[3:0] Call Progress Tone TC (time constant) determines the response of the Call Progress Tone energy averaging. Larger values imply a larger time constant, greater stability but less responsive to changes in energy level.

CPT_EN Enable the Call Progress Tone detection block

8.16.2 CPT_STATUS

Address	Access Mode	Value At Reset	Nominal Value
0x281	R	0x00	

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	CPT_RX

CPT_RX Call Progress Tone Detected. 1: Energy exceeds CPT_THRES_H 0: Energy less than CPT_THRES_L or CPT_EN=0.

8.16.3 CPT_THRES_H

Address	Access Mode	Value At Reset	Nominal Value
0x282~0x283	R/W	0x2400	

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
82	CPT_THRES_H [15:8]							
83	CPT_THRES_H [7:0]							

CPT_THRES_H[15:0] This register defines the on/high threshold of the Call Progress Tone detection. If the energy is greater than CPT_THRES_H, CPT_RX will be set. The 16 bit value represents the decimal fraction of full scale in 2's complement format.

8.16.4 CPT_THRES_L

Address	Access Mode	Value At Reset	Nominal Value
0x284~0x285	R/W	0x1A00	

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
84	CPT_THRES_L [15:8]							
85	CPT_THRES_L [7:0]							

CPT_THRES_L[15:0] This register defines the off/low threshold of the Call Progress Tone detection. If the energy is less than CPT_THRES_L, CPT_RX will be cleared. The 16 bit value represents the decimal fraction of full scale in 2's complement format.

8.16.5 CPT_ENERGY

Address	Access Mode	Value At Reset	Nominal Value
0x286~0x287	R	0x0000	

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
86	CPT_ENERGY [15:8]							
87	CPT_ENERGY [7:0]							

CPT_ENERGY[15:0] This register represents the instantaneous signal power from 300Hz to 650Hz. The value is in the range, 0x0000 to 0xFFFF, with larger value representing larger signal power.

8.17 Ring Detection and Pulse / Period Width Measurement

Schmitt trigger:

The Schmitt trigger input to the ISD61S00 has the nominal characteristics of VH=1.63 volt and VL=1.33 volt.

Pulse / Period Width Measurement (PPM):

The interrupt occurrence could be selected to occur at falling edge or rising edge or both edges of ringing signal. For example, if the rising edge has been selected, an interrupt event will be sent at each rising edge of ringing signal. The RNG_CTRL[1:0] is used to select the interrupt occurrence. Figure 8-26 shows the function block of ring detector and Figure 8-27 illustrates the operation of ring detector by the timing waveforms. The ringing signal goes through the Schmitt trigger. The PPM interrupt occurrence register RINGIntrptSel[1:0] is set to 0x03 in the illustration, so the RINGIN signal is then examined by the PPM interrupt generator to generate the interrupt event at the both edge transitions of RINGIN signal. To save the software computation power, PPM counter logic can report the length between consecutive interrupt events by a counter value. The counter value is stored in register RINGCNTR that is related to the PPM counter operating frequency. Depending on the time resolution to measure the length between consecutive interrupt events.

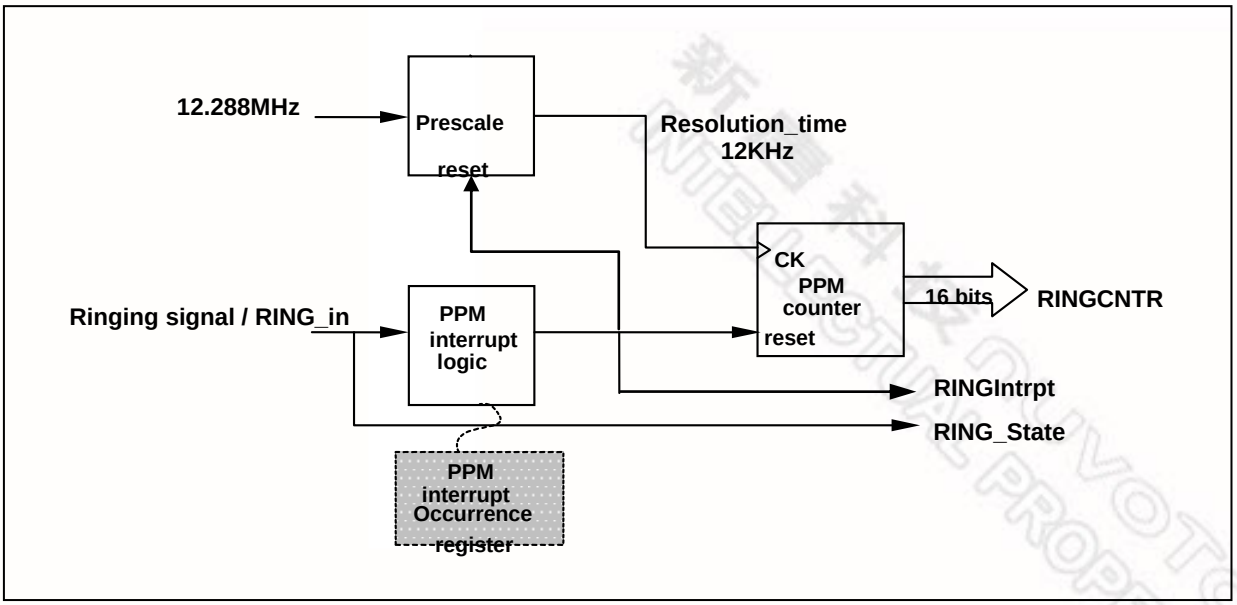


Figure 8-26 Block diagram of ring detector

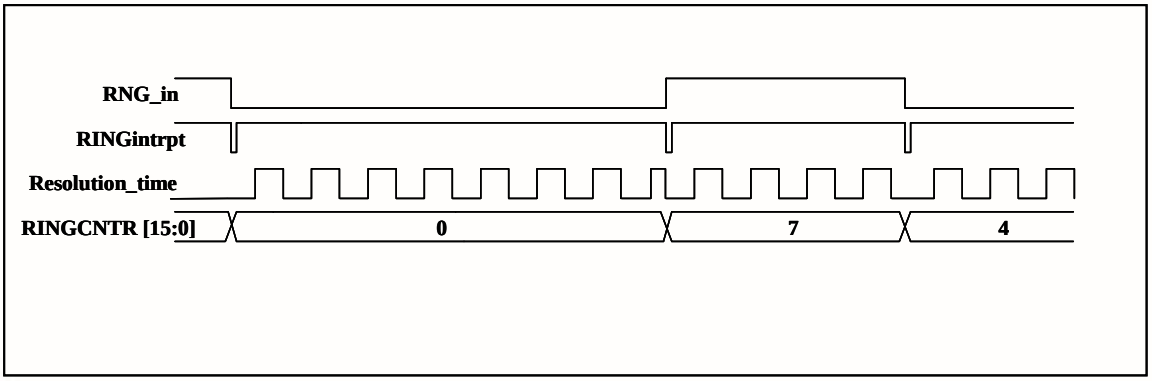


Figure 8-27 The timing of ring detector with RNG_CTRL[1:0]=2'b11

8.17.1 RNG_CTRL

Address	Access Mode	Value At Reset	Nominal Value
0x01B0	R/W	0x00	

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
RING_EN	RESERVED	RESERVED	RESERVED	RES[1]	RES[0]	INT_SEL[1]	INT_SEL[0]

RING_EN RING Operation Enable 1=Enable, 0=Disable

RES[1:0] Select the resolution of the PPM measurement. The maximum time interval measureable is 65536 * resolution time unit. See Table 8-19.

INT_SEL[0] =1 Generates an interrupt on falling edge.

INT_SEL[1] =1 Generates an interrupt on rising edge.

Interrupt Occurrence	Address=0x01B0	
	Bit[1]	Bit[0]
No Interrupt	0	0
Interrupt on falling edge	0	1
Interrupt on rising edge	1	0
Interrupt on both falling and rising edge	1	1

Table 8-19 Values of PPM Timer resolution.

RES[1:0]	Resolution Time Unit
0	1/12KHz = 83us
1	1/24 KHz = 41.7us
2	1/48 KHz = 20.8us
3	1/96 KHz = 10.4us

8.17.2 RNG_STATE

Address	Access Mode	Value At Reset	Nominal Value
0x01B1	R	0x00	

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RING_STATE

RING_STATE This bit reflects the state of ringing signal passing through the Schmitt trigger input pad.

8.17.3 RNG_CNTR

Address	Access Mode	Value At Reset	Nominal Value
0x1B2~0x1B3	R	0x0000	

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
B2	RNG_CNTR[15:8]							
B3	RNG_CNTR[7:0]							

The RNG_CNTR register is updated with the PPM counter value as the interrupt is generated. The PPM counter is a 16-bit counter which is reset to 0 and starts to increase by 1 to maximum value 65535, when the PPM interrupt is generated. The PPM counter will saturate at the maximum value 65536 until the next interrupt occurrence.

Table 8-19 illustrates the expected counter values for detecting ringing signals according to the requirements of the BT and North American systems.

Table: 8-16 PPM counter value vs. PPM resolution for various ringing frequencies and cadences.

PPM counter operating frequency (Hz)	PPM counter resolution	USA Freq								BT Cadence		
		10 Hz	20Hz	15.8 Hz	50Hz	68 Hz	69 Hz	70 Hz				
		100 ms	50 ms	63.29 ms	20 ms	14.7 ms	14.49 ms	14.28 ms	4 sec off	0.2 sec off	0.4 sec on	2 sec off
12K	0.083 ms	1200 (04B0)	600 (0258)	759 (02F7)	240 (00F0)	176 (00B0)	173 (00AD)	171 (00AB)	48200	2410	4820	24100
24K	0.042ms	2400 (0960	1200 (04B0)	1518 (05EE)	480 (01E0)	352 (0160)	348 (15C)	342 (0156)	65535	4762	9524	47620
48K	0.021ms	4800 (12C0)	2400 (0960)	3038 (0BDE)	960 (3C0)	706 (2C2)	696 (2B8)	686 (2AE)	65535	9524	19047	65535
96K	0.01ms	9600 (2580)	4800 (12C0)	6076 (17BC)	1920 (0780)	1412 (0584)	1392 (0570)	1372 (055C)	65535	20000	40000	65535

PPM counter, max (16 bit counter)=65535 (PPM counter value= PPM counter operating frequency / Frequency)

8.17.4 RNG_LATCH

Address	Access Mode	Value At Reset	Nominal Value
0x1B4	R/W	0x00	

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	LATCH

LATCH

0: RNG_CNTR is updated each interrupt occurrence regardless of whether interrupt has been serviced.

1: The RNG_CNTR value is latched until RNG_LATCH is written again. This prevents noise on ring signal or slow interrupt servicing causing missed measurements. With LATCH=1, to service interrupt read RNG_CNTR, then write RNG_LATCH. Once written, next interrupt will update RNG_CNTR to a new value. If RNG_LATCH is not written, new interrupts will continue to be generated but RNG_CNTR will not be updated.

8.17.5

8.18 Timer

The ISD61S00 incorporates a programmable 1-millisecond resolution timer. It can be programmed to generate a pulse in the range 1 ~ 65536 milliseconds (1 millisecond spacing). When the timer is enabled by writing to TIME_CTRL[7]=1 the 1 ms timer will start to count until the current timer count (TIME_CNT) is equal to the target count (TIME_TARG) at which time an interrupt event is generated. A write of the timer control register (TIME_CTRL) will force timer to reset/clear. The interval between timer interrupts will be TIME_TARG+1ms.

8.18.1

8.18.2 TIME_TARG

Address	Access Mode	Value At Reset	Nominal Value
0x292~0x293	R/W	0x0000	

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
92	TIME_TARG[15:8]							
93	TIME_TARG[7:0]							

TIME_TARG[15:0] TIME_TARG+1 ms timer target count, maximum count 65536 ms.

8.18.3 TIME_CNT

Address	Access Mode	Value At Reset	Nominal Value
0x294~0x295	R	0x0000	

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
94	TIME_CNT[15:8]							
95	TIME_CNT[7:0]							

TIME_CNT[15:0] Current timer count number. Unit: ms.

8.19 Gain Stage and Mixer

The Gain stage and Mixer block allows routing of PCM signals between various blocks. There are five gain- mixer channels allowing mixes for:

1. AEC Input (to air CODEC)
2. LEC Input (to PSTN CODEC)
3. Record (to audio compression block).
4. I2S Left Channel.
5. I2S Right Channel.

Each of the mixers has five variable gain inputs along with inputs from the tone generator and FSK encoder. The five variable gain inputs are:

1. MIC – output from air CODEC.
2. PLAY – output from audio decompression.
3. TI – output from PSTN CODEC.
4. I2S_L – input from I2S left channel.
5. I2S_R – input from I2S right channel.

The block diagram of a single mixer channel is shown in Figure 8-28.

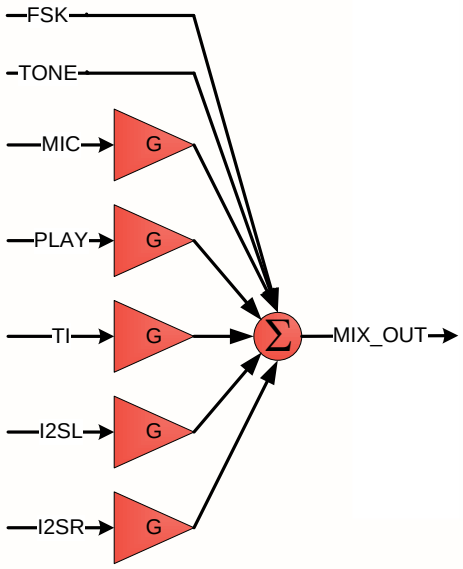


Figure 8-28 Gain/Mixer Channel

In addition to the mixers there are also gain stages associated with each of the CODECs and echo cancellers that allow input and output gain control along with side tone insertion.

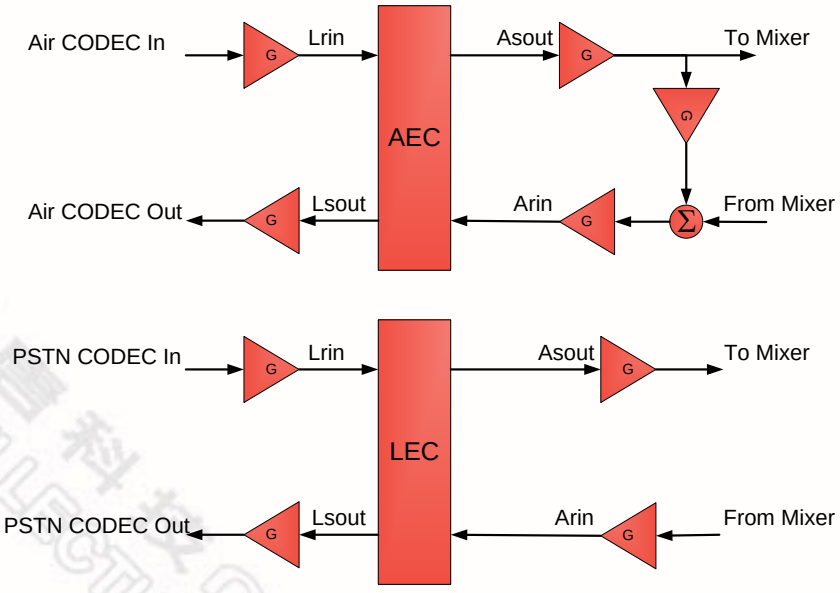


Figure 8-29 CODEC/EC Gain Stages

Each gain stage is controlled by seven bits of a configuration register. The gain is configurable from +24dB to -31.5dB in 0.5dB steps. The gain is a seven bit two's complement number with 0x40

corresponding to a total MUTE of the signal as shown in Table 8-20. The list of mixer gain registers and their associated path is given in Table 8-21.

Table 8-20 Gain Stage Control value Mapping.

Index	Gain (dB)	Index	Gain (dB)	Index	Gain (dB)	Index	Gain (dB)
0x00	0.0	0x20	16.0	0x7F	-0.5	0x5F	-16.5
0x01	0.5	0x21	16.5	0x7E	-1.0	0x5E	-17.0
0x02	1.0	0x22	17.0	0x7D	-1.5	0x5D	-17.5
0x03	1.5	0x23	17.5	0x7C	-2.0	0x5C	-18.0
0x04	2.0	0x24	18.0	0x7B	-2.5	0x5B	-18.5
0x05	2.5	0x25	18.5	0x7A	-3.0	0x5A	-19.0
0x06	3.0	0x26	19.0	0x79	-3.5	0x59	-19.5
0x07	3.5	0x27	19.5	0x78	-4.0	0x58	-20.0
0x08	4.0	0x28	20.0	0x77	-4.5	0x57	-20.5
0x09	4.5	0x29	20.5	0x76	-5.0	0x56	-21.0
0x0A	5.0	0x2A	21.0	0x75	-5.5	0x55	-21.5
0x0B	5.5	0x2B	21.5	0x74	-6.0	0x54	-22.0
0x0C	6.0	0x2C	22.0	0x73	-6.5	0x53	-22.5
0x0D	6.5	0x2D	22.5	0x72	-7.0	0x52	-23.0
0x0E	7.0	0x2E	23.0	0x71	-7.5	0x51	-23.5
0x0F	7.5	0x2F	23.5	0x70	-8.0	0x50	-24.0
0x10	8.0	0x30	24.0	0x6F	-8.5	0x4F	-24.5
0x11	8.5	0x31	24.0	0x6E	-9.0	0x4E	-25.0
0x12	9.0	0x32	24.0	0x6D	-9.5	0x4D	-25.5
0x13	9.5	0x33	24.0	0x6C	-10.0	0x4C	-26.0
0x14	10.0	0x34	24.0	0x6B	-10.5	0x4B	-26.5
0x15	10.5	0x35	24.0	0x6A	-11.0	0x4A	-27.0
0x16	11.0	0x36	24.0	0x69	-11.5	0x49	-27.5
0x17	11.5	0x37	24.0	0x68	-12.0	0x48	-28.0
0x18	12.0	0x38	24.0	0x67	-12.5	0x47	-28.5
0x19	12.5	0x39	24.0	0x66	-13.0	0x46	-29.0
0x1A	13.0	0x3A	24.0	0x65	-13.5	0x45	-29.5
0x1B	13.5	0x3B	24.0	0x64	-14.0	0x44	-30.0
0x1C	14.0	0x3C	24.0	0x63	-14.5	0x43	-30.5
0x1D	14.5	0x3D	24.0	0x62	-15.0	0x42	-31.0
0x1E	15.0	0x3E	24.0	0x61	-15.5	0x41	-31.5
0x1F	15.5	0x3F	24.0	0x60	-16.0	0x40	**Mute**

The registers that control each gain stage are listed in Table 8-21 and their position is indicated by their address in Table 8-21.

Table 8-21 Mixer Gain Registers.

Register	Description	Address
GS_ACST	Air CODEC Side tone	111
GS_ACIG	Air CODEC input Gain	112
GS_ACOG	Air CODEC output Gain	113
GS_AEOG	AEC Output Gain	114
GS_AEIG	AEC Input Gain	115
GS_LCIG	Line CODEC input Gain	116
GS_LCOG	Line CODEC output Gain	117
GS_LEOG	LEC Output Gain	118
GS_LEIG	LEC Input Gain	119
GS_AOAI	AEC out - AEC in Gain	11A
GS_LOAI	LEC out - AEC in Gain	11B
GS_PLAI	PLAY - AEC in Gain	11C
GS_ILAI	I2SL - AEC in	11D
GS_IRAI	I2SR - AEC in	11E
GS_AOLI	AEC out - LEC in Gain	11F
GS_LOLI	LEC out - LEC in Gain	120
GS_PLLI	PLAY - LEC in Gain	121

Register	Description	Address
GS_ILLI	I2SL - LEC in	122
GS_IRLI	I2SR - LEC in	123
GS_AORI	AEC out - REC in Gain	124
GS_LORI	LEC out - REC in Gain	125
GS_PLRI	PLAY - REC in Gain	126
GS_ILRI	I2SL - REC in	127
GS_IRRI	I2SR - REC in	128
GS_AOIL	AEC out - I2SL in Gain	129
GS_LOIL	LEC out - I2SL in Gain	12A
GS_PLIL	PLAY - I2SL in Gain	12B
GS_ILIL	I2SL - I2SL in	12C
GS_IRIL	I2SR - I2SL in	12D
GS_AOIR	AEC out - I2SR in Gain	12E
GS_LOIR	LEC out - I2SR in Gain	12F
GS_PLIR	PLAY - I2SR in Gain	130
GS_ILIR	I2SL - I2SR in	131
GS_IRIR	I2SR - I2SR in	132

新唐科技 NUVOTON INTELLECTUAL PROPERTY

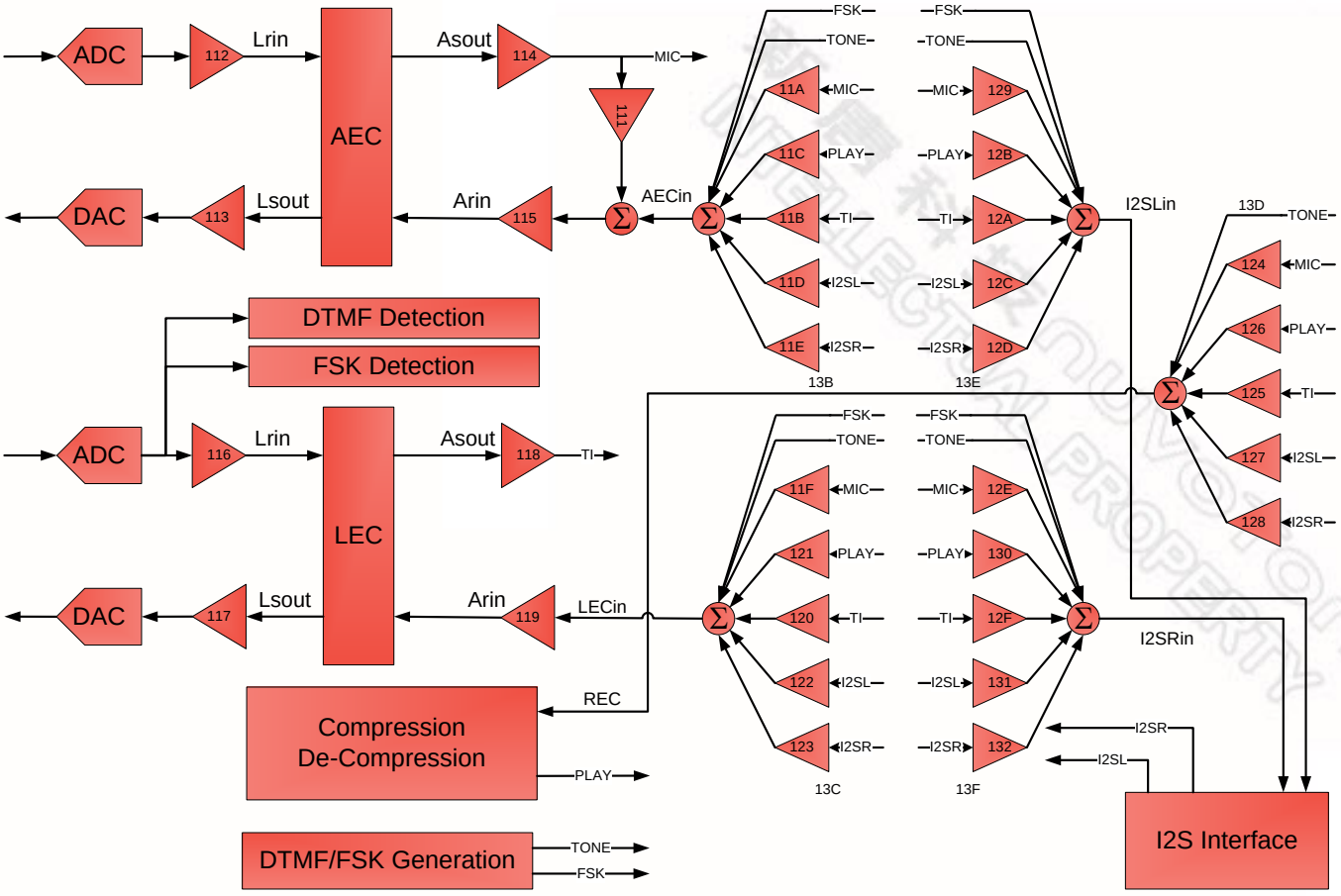


Figure 8-30 Gain Stages and Mixer with Control Addresses

8.19.1 GS_CTRL

Address	Access Mode	Value At Reset	Nominal Value
0x110	R/W	0x00	

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
GSMIX_EN	-						

GSMIX_EN Enable the Mixer Gain Stage.

8.19.2 CODEC EC Gain Stages

Address	Access Mode	Value At Reset	Nominal Value
0x111-0x119	R/W	0x00	

Address	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x111	POSITION	G1: Air CODEC Side-Tone Gain [6:0]						
0x112	Reserved	G2: Air ADC To AEC Gain [6:0]						
0x113	Reserved	G3: AEC To Air DAC Gain [6:0]						
0x114	Reserved	G4: AEC To MIC Gain [6:0]						
0x115	Reserved	G5: AECin Mixer To AEC Gain [6:0]						
0x116	Reserved	G6: Line ADC To LEC Gain [6:0]						
0x117	Reserved	G7: LEC To Line DAC Gain [6:0]						
0x118	Reserved	G8: LEC To TI Gain [6:0]						
0x119	Reserved	G9: LECin Mixer To LEC Gain [6:0]						

Air CODEC Side-Tone Gain [6:0]	Side-tone gain (2's complement). --- Operation range: -0.5 dB ~ -31.5 dB, 0.5 dB per step. --- 0x40 means mute/disable this path.
POSITION	Determines whether the side tone insertion is before or after the echo cancellation block. 0: After AEC as per Figure 8-30. 1: Before – immediately after DAC/ADC.
Air ADC To AEC Gain [6:0] AEC To Air DAC Gain [6:0] AEC To MIC Gain [6:0] AEC To MIC Gain [6:0] CODEC To LEC Gain [6:0] LEC To CODEC Gain [6:0] LEC To Mixer Gain [6:0] Mixer To LEC Gain [6:0]	Gain in 2's complement format, G*0.5dB. Operation range: +24 dB ~ -31.5 dB, 0.5 dB per step. 0x40 mutes gain stage. Note: As per Table 8-20

8.19.3 AECin Path Mixing Gain Control

Address	Access Mode	Value At Reset	Nominal Value
0x11A-0x11E	R/W	0x00	

Address	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x11A	Reserved	GA: AEC to AECin Path Mixer Gain Index [6:0]						
0x11B	Reserved	GB: LEC to AECin Path Mixer Gain Index [6:0]						
0x11C	Reserved	GC: PLAY to AECin Path Mixer Gain Index [6:0]						
0x11D	Reserved	GD: I2S L to AECin Path Mixer Gain Index [6:0]						
0x11E	Reserved	GE: I2S R to AECin Path Mixer Gain Index [6:0]						

AEC to AECin Path Mixer Gain Index [6:0]
LEC to AECin Path Mixer Gain Index [6:0]
PLAY to AECin Path Gain Index [6:0]
I2S L to AECin Path Mixer Gain Index [6:0]
I2S R to AECin Path Mixer Gain Index [6:0]

Gain in 2's complement format, G*0.5dB.
- Operation range: +24 dB ~ -31.5 dB, 0.5 dB per step.
- 0x40 mutes gain stage.

Note: As per Table 8-20

8.19.4 LECin Path Mixing Gain Control

Address	Access Mode	Value At Reset	Nominal Value
0x11F-0x123	R/W	0x00	

Address	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x11F	Reserved	GF : AEC to LECin Path Mixer Gain Index [6:0]						
0x120	Reserved	G10: LEC to LECin Path Mixer Gain Index [6:0]						
0x121	Reserved	G11: PLAY to LECin Path Mixer Gain Index [6:0]						
0x122	Reserved	G12: I2S L to LECin Path Mixer Gain Index [6:0]						
0x123	Reserved	G13: I2S R to LECin Path Mixer Gain Index [6:0]						

AEC to LECin Path Mixer Gain Index [6:0]
LEC to LECin Path Mixer Gain Index [6:0]

Gain in 2's complement format, G*0.5dB.
- Operation range: +24 dB ~ -31.5 dB, 0.5 dB per step.
- 0x40 mutes gain stage.

PLAY to LECin Path Gain Index [6:0]
I2S L to LECin Path Mixer Gain Index [6:0]
I2S R to LECin Path Mixer Gain Index [6:0]

Note: As per Table 8-20

8.19.5 REC Path Mixing Gain Control Registers

Address	Access Mode	Value At Reset	Nominal Value
0x124-0x128	R/W	0x00	

Address	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x124	Reserved	G14 : AEC to REC Path Mixer Gain Index [6:0]						
0x125	Reserved	G15: LEC to REC Path Mixer Gain Index [6:0]						
0x126	Reserved	G16: PLAY to REC Path Mixer Gain Index [6:0]						
0x127	Reserved	G17: I2S L to REC Path Mixer Gain Index [6:0]						
0x128	Reserved	G18: I2S R to REC Path Mixer Gain Index [6:0]						

AEC to REC Path Mixer Gain Index [6:0]
LEC to REC Path Mixer Gain Index [6:0]
PLAY to REC Path Gain Index [6:0]
I2S L to REC Path Mixer Gain Index [6:0]
I2S R to REC Path Mixer Gain Index [6:0]

Gain in 2's complement format, G*0.5dB.

- Operation range: +24 dB ~ -31.5 dB, 0.5 dB per step.
- 0x40 mutes gain stage.

Note: As per Table 8-20

新唐科技 NUVOTON INTELLECTUAL PROPERTY

8.19.6 I2SLin Path Mixing Gain Control Registers

Address	Access Mode	Value At Reset	Nominal Value
0x129-0x12D	R/W	0x00	

Address	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x129	Reserved	G19 : AEC to I2SLin Path Mixer Gain Index [6:0]						
0x12A	Reserved	G1A: LEC to I2SLin Path Mixer Gain Index [6:0]						
0x12B	Reserved	G1B: PLAY to I2SLin Path Mixer Gain Index [6:0]						
0x12C	Reserved	G1C: I2S L to I2SLin Path Mixer Gain Index [6:0]						
0x12D	Reserved	G1D: I2S R to I2SLin Path Mixer Gain Index [6:0]						

AEC to I2SLin Path Mixer Gain Index [6:0]
LEC to I2SLin Path Mixer Gain Index [6:0]
PLAY to I2SLin Path Gain Index [6:0]
I2S L to I2SLin Path Mixer Gain Index [6:0]
I2S R to I2SLin Path Mixer Gain Index [6:0]

Gain in 2's complement format, G*0.5dB.

- Operation range: +24 dB ~ -31.5 dB, 0.5 dB per step.
- 0x40 mutes gain stage.

Note: As per Table 8-20

8.19.7 I2SRin Path Mixing Gain Control Registers

Address	Access Mode	Value At Reset	Nominal Value
0x12E-0x132	R/W	0x00	

Address	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x12E	Reserved	G1E : AEC to I2SRin Path Mixer Gain Index [6:0]						
0x12F	Reserved	G1F: LEC to I2SRin Path Mixer Gain Index [6:0]						
0x130	Reserved	G20: PLAY to I2SRin Path Mixer Gain Index [6:0]						
0x131	Reserved	G21: I2S L to I2SRin Path Mixer Gain Index [6:0]						
0x132	Reserved	G22: I2S R to I2SRin Path Mixer Gain Index [6:0]						

AEC to I2SRin Path Mixer Gain Index [6:0]
LEC to I2SRin Path Mixer Gain Index [6:0]
PLAY to I2SRin Path Gain Index [6:0]
I2S L to I2SRin Path Mixer Gain Index [6:0]
I2S R to I2SRin Path Mixer Gain Index [6:0]

Gain in 2's complement format, G*0.5dB.

- Operation range: +24 dB ~ -31.5 dB, 0.5 dB per step.
- 0x40 mutes gain stage.

Note: As per Table 8-20

8.19.8 Mixer Source Enable Registers

Address	Access Mode	Value At Reset	Nominal Value
0x13B-0x13F	R/W	0x00	

Address	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	Reserved	FSK_GEN	TONE_GEN	I2S_R	I2S_L	I15K	TI	MIC
0x13B	Reserved	AECin Mixer Source Enable [6:0]						
0x13C	Reserved	LECin Mixer Source Enable [6:0]						
0x13D	Reserved	REC Mixer Source Enable [6:0]						
0x13E	Reserved	I2SLin Mixer Source Enable [6:0]						
0x13F	Reserved	I2SRin Mixer Source Enable [6:0]						

AECin Mixer Source Enable [6:0]
LECin Mixer Source Enable [6:0]
REC Mixer Source Enable [6:0]
I2SLin Mixer Source Enable [6:0]
I2SRin Mixer Source Enable [6:0]

These registers determine which of the input sources to the mixers are enabled.
1 = Source Enable, 0 = Source Disable
For example if bit 6 of AECin register (0x13B) is set then output of the FSK generator will be mixed to the output. If bit 1 is set then the TI input will be mixed to the output.

Note: The Mixing operation is performed after Mixer Gain.

Mixer Example:

--- 0x 013B = 0x 01,AEC out = AEC in *GA **(Loop Back)**

--- 0x 013C = 0x 12, LEC out = I2S_R in *G12 **+ LEC in**G10

--- 0x 013D = 0x 23, I15K out = DTMF_GEN in + LEC in *G15 **+ AEC in** *G14

--- 0x 013E = 0x 37, I2S_L out = DTMF_GEN in + I2S_R in *G1D **+ I15K in** *G1B **+ LEC in** *G1A **+ AEC in** *G19

--- 0x 013F = 0x 7F, I2S_R out = FSK_GEN in + TONE_GEN in + I2S_R in *G22 **+**
I2S_L in *G21**+ I15K in** *G20 **+ LEC in** *G1F **+ AEC in** *G1E

8.20 Air and Line CODEC

The ISD61S00 contains two channels of A/D and D/A conversion for the air side interface and the line side interface.

8.20.1 AC_EN, LC_EN – Air /Line CODEC Enable Register

Address	Access Mode	Value At Reset
0x140 (Air) 0x150 (Line)	R/W	0x20

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CODEC_EN	Reserved	IMAGE_FIL	SAT_ON	HPF_FREQ [3:2]		-	

CODEC_EN	CODEC Digital Path Enable. 1 = Enable, 0 = Disable (Power Down).
IMAGE_FIL	Enable image filter. The image IIR filter may be used to remove aliased image of signal resulting from down-sampling. This filter is available for use in 8kHz mode on the line side and/or the air side, and in 16kHz mode on the air side only.
SAT_ON	Enable MACC saturation control. It is recommended that this is enabled.
HPF_FREQ [3:2]	Control for high pass filter cut-off frequency. For 16kHz sampling the cut-off frequencies are: 00 = 6.7Hz Cutoff 01 = 81.7Hz Cutoff 10 = 204Hz Cutoff 11 = Reserved. For 8kHz sampling the cut-off frequencies are: 00 = 3.3Hz Cutoff 01 = 40.8Hz Cutoff 10 = 102Hz Cutoff 11 = Reserved.

To ensure the correct reset and initial conditions of the CODEC the following sequence should be observed for power up and down of the CODEC.

CODEC power up sequence:

1. Begin at state 0x00 (CODEC_EN=0, IMAGE_FIL=0, SAT_ON=0)
2. Enable CODEC 0x80 (CODEC_EN=1, IMAGE_FIL=0, SAT_ON=0)
3. Enable image filter (if applicable) 0xB0 (CODEC_EN=1, IMAGE_FIL=1, SAT_ON=1)

CODEC power down sequence:

- Set to state 0x00 (CODEC_EN=0, IMAGE_FIL=0, SAT_ON=0).

8.20.2 AC_CTRL, LC_CTRL – Air /Line CODEC Dither Control

Address	Access Mode	Value At Reset
0x141 (Air) 0x151(Line)	R/W	0x07

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Reserved	Reserved	Reserved		Reserved	Dither_sign	Dither_level	Dither_enable

Dither_enable = 1: Enable dither;

Dither_level = 1: 17bits dither; 0: 16bits dither;

Dither_sign =0: Additional random noise is both positive and negative.
=1: Additional random noise is only positive.

8.20.3 AC_ADCG, LC_ADCG – Air/Line CODEC ADC Gain.

Address	Access Mode	Value At Reset	Nominal Value
0x142~0x143 (Air) 0x152~0x153 (Line)	R/W	0x0400	

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
2					ADC_DIG_GAIN[11:8]			
3	ADC_DIG_GAIN[7:0]							

ADC_DIG_GAIN [11:0] Digital ADC Path Gain. (Default = 12'h400, unity gain)
Format: 2's complements number with format = 2.10
Example:
--- ADC_DIG_GAIN [11:0] = 12'h5A6, Gain = + 3 dB
--- ADC_DIG_GAIN [11:0] = 12'h400, Gain = 0 dB
--- ADC_DIG_GAIN [11:0] = 12'h2D6, Gain = - 3 dB

8.20.4 AC_DACG, LC_DACG – Air/Line CODEC DAC Gain.

Address	Access Mode	Value At Reset	Nominal Value
0x144~0x145 (Air) 0x154~0x155 (Line)	R/W	0x0400	

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
4					DAC_DIG_GAIN[11:8]			
5	DAC_DIG_GAIN[7:0]							

DAC_DIG_GAIN [11:0]	Digital DAC Path Gain. (Default = 12'h400, unity gain) Format: 2's complements number with format = 2.10 Example: --- DAC_DIG_GAIN [11:0] = 12'h5A6, Gain = + 3 dB --- DAC_DIG_GAIN [11:0] = 12'h400, Gain = 0 dB --- DAC_DIG_GAIN [11:0] = 12'h2D6, Gain = - 3 dB

8.20.5

8.20.6

8.21 RINGER TONE GENERATOR

The ringer tone will be generated while an incoming call occurred. There are two tone signals can be mixed to the speakerphone driver output. This subsection describes the Ringer Tone Generator with the PWM (Pulse Width Modulation) format.

Ringer tone generator (PWM) specification:

- Tone Channel Number = 2
- Tone Volume Step = 32
- Tone Frequency Range = 93Hz~24KHz

$$Frequency = \frac{12.288M}{16 \times N \times 32}\ Hz \qquad N = 1 \sim 256$$

The tone frequency/volume control signal path is shown as Figure 8-31 Ringer Tone Generator Block

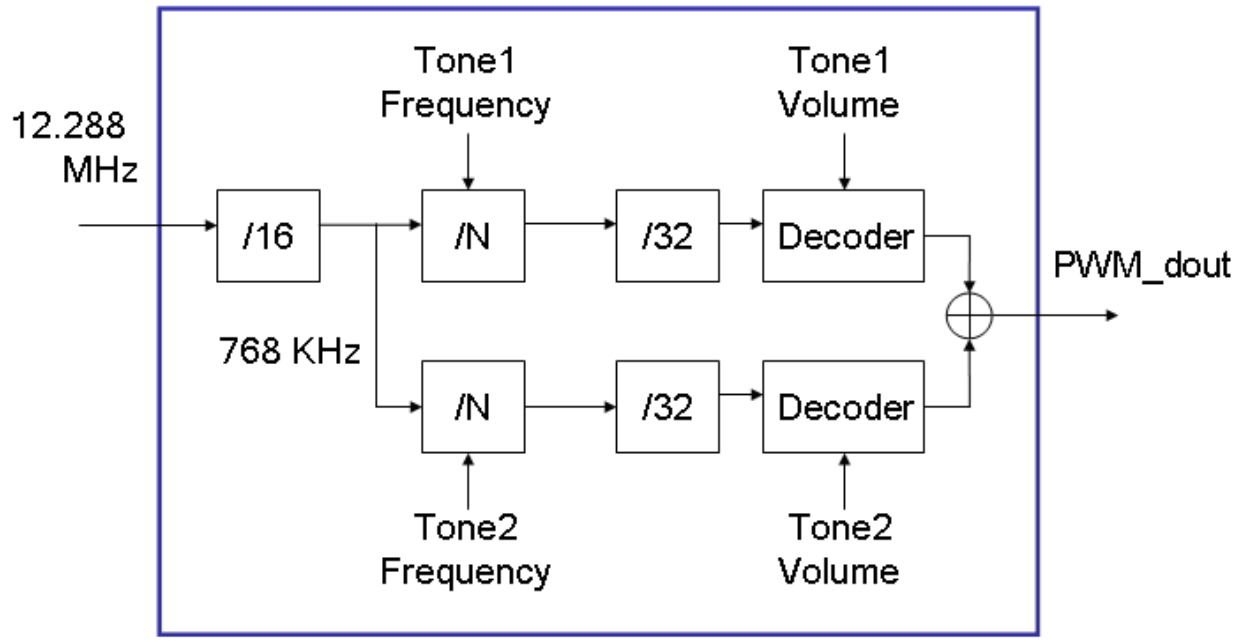


Figure 8-31 Ringer Tone Generator Block

8.21.1 PWM Clock

Address	Access Mode	Value At Reset	Nominal Value
0x1A0	R/W	0x00	

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PWM CLK ON	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved

PWM CLK ON — PWM Operation Clock Enable. 1 = Enable 0 = Disable

--- Set this bit will enable PWM operation clock.

Note: Before disable this bit, the tone_on1 (0x1A2: BIT [7]) and tone_on2 (0x1A4 BIT [7]) must be disable first.

8.21.2 PWM Tone1 Control

Address	Access Mode	Value At Reset	Nominal Value
0x1A2	R/W	0x00	

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TONE ON 1	Reserved	Reserved	TONE_VOLUME_1 [4:0]				

TONE_VOLUME_1 [4:0] TONE 1 Volume setting.
--- There have 32 stage volume of the Tone generator.
--- Set this volume zero will force output mute.

TONE ON 1 TONE 1 Enable. 1 = Enable, 0 = Disable
Note: This bit can be updated only while PWM CLK ON (0x1A0 BIT [7]) is active. Besides, if the Ringer what to on/off with a full frequency cycle, the suggest way is to play one “Zero” volume cycle before disable the TONE ON.

8.21.3 PWM Tone1 Frequency

Address	Access Mode	Value At Reset	Nominal Value
0x1A3	R/W	0x00	

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TONE_FREQ_1 [7:0]							

TONE_FREQ_1 [7:0] TONE 1 Frequency setting.
--The Ringer Tone Frequency = 768 / [32*(TONE_FREQ_1+ 1)] KHz
EX: If the TONE_FREQ_1 = 0x4A, then the Ringer Tone 1 output frequency will be 768 / [32 * (74+1)] = 320 Hz

8.21.4 PWM Tone2 Control

Address	Access Mode	Value At Reset	Nominal Value
0x1A4	R/W	0x00	

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TONE ON 2	Reserved	Reserved	TONE_VOLUME_2 [4:0]				

TONE_VOLUME_2 [4:0] TONE 2 Volume setting.
--- There have 32 stage volume of the Tone generator.
--- Set this volume zero will force output mute.

TONE ON 2	TONE 2 Enable. 1 = Enable, 0 = Disable Note: This bit can be updated only while PWM CLK ON (0x1A0 BIT [7]) is active. Besides, if the Ringer what to on/off with a full frequency cycle, the suggest way is to play one "Zero" volume cycle before disable the TONE ON.

8.21.5 PWM Tone2 Frequency

Address	Access Mode	Value At Reset	Nominal Value
0x1A5	R/W	0x00	

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TONE_FREQ_2 [7:0]							

TONE_FREQ_2 [7:0]	TONE 2 Frequency setting. ---The Ringer Tone Frequency = 750/[32*(TONE_FREQ_2 + 1)] KHz EX: If the TONE_FREQ_2 = 0x4A, then the Ringer Tone 2 output frequency will be 768 /[32 * (74+1)] = 320 Hz

9. ACOUSTIC PROCESSING BLOCK

The Acoustic possessing block performs a variety of functions to implement the full or half duplex echo cancellation. The ISD61S00 incorporates an Acoustic Echo Cancellation (AEC) block for the microphone and speaker CODEC and a Line Echo Cancellation (LEC) block for the PSTN interface. These blocks are very similar, with the main differences being that the AEC includes an AGC function and has a greater echo cancellation length.

The functions performed by the acoustic processing blocks include:

Echo Cancellation: This is an adaptive estimation of the characteristics of the echo path to cancel echo introduced by speaker/microphone acoustic coupling and echo return paths from the environment.

Voice Detection (VD): Decides whether input signal from the far end is an active voice signal by estimating the signal power.

Double Talk (DT) Detection: Decides whether input signal from the near end is an active voice signal by estimating the signal power.

AGC: Automatic gain control to provide an optimal level from the microphone input.

Soft Clipping: limits the input signal path to prevent hard saturation of the output.

Acoustic Suppression: Used in half duplex to suppress voice either from the near end or the far end.

Noise Suppression: Detects and suppresses noise.

9.1 Full/Half AEC Block Diagram

Full acoustic echo cancellation unit removes the echo signal inserted by the speaker coupling and space reflections. Half duplex AEC smoothly transitions the acoustic suppressors from one direction to the other based on the power estimations of the voice and double talk detectors. Figure 9-1 illustrates the block diagram of the Full/Half Acoustic Echo Canceller. The line echo cancellation block is similar though has the AGC block replaced by an additional soft clipping gain block (Figure 9-2).

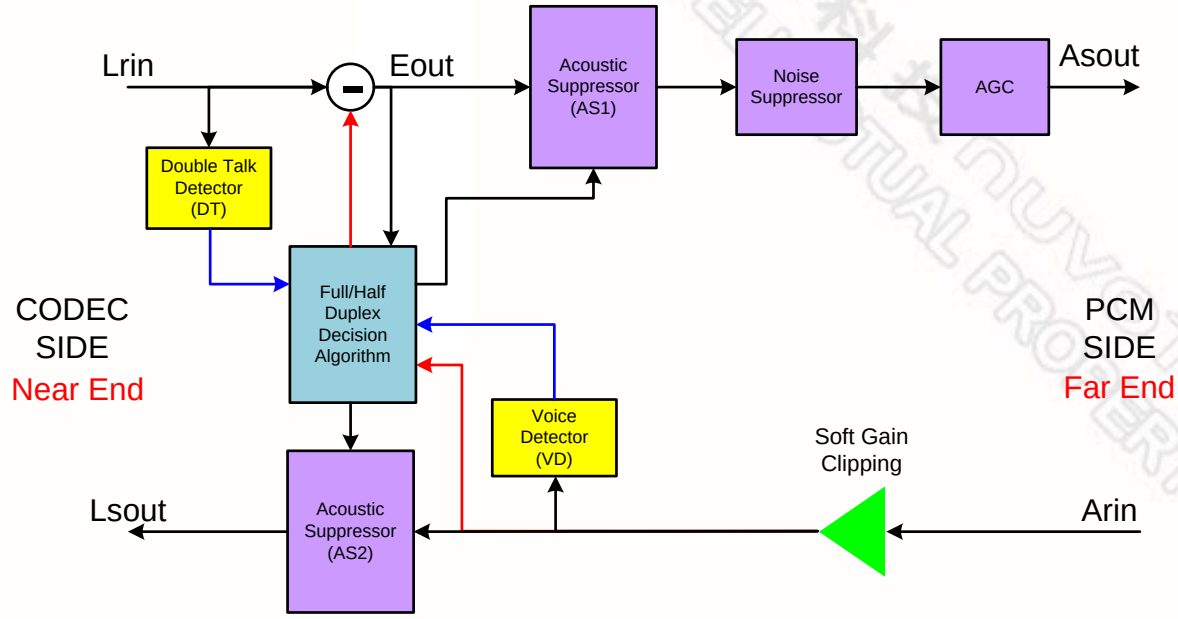


Figure 9-1: Signal flow through the Acoustic Echo Canceller (AEC) in the speech Processor.

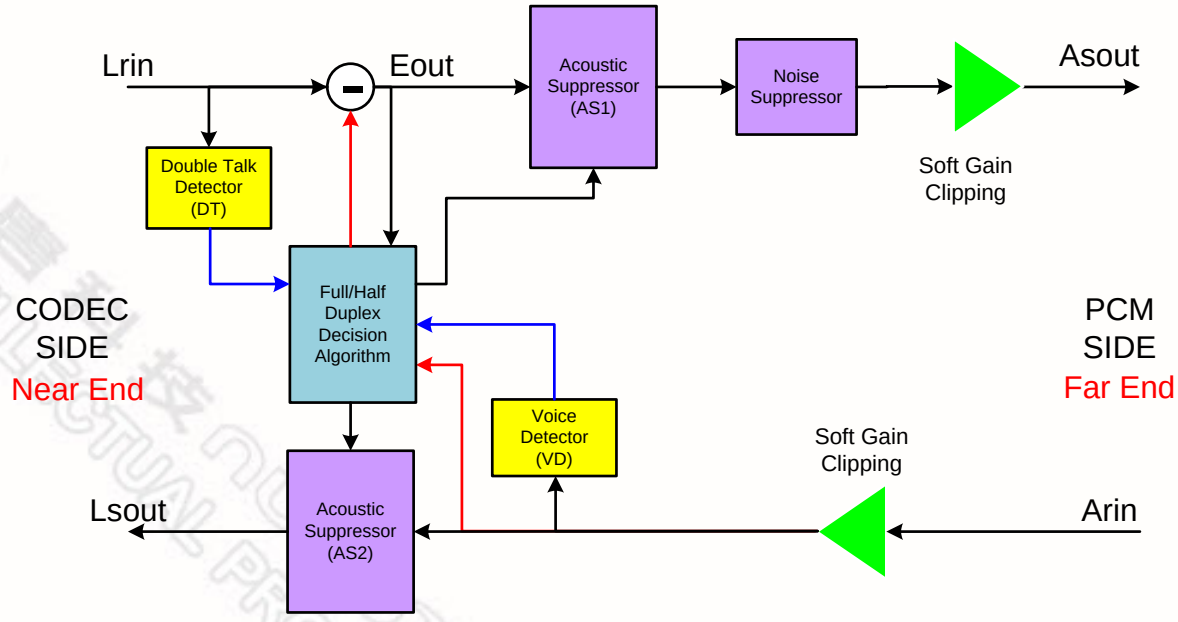


Figure 9-2: Signal flow through the Line Echo Canceller (LEC) in the speech Processor.

9.2 Control Register Memory Map

The AEC/LEC block is controlled via configuration registers from that enable control and tuning of the echo cancellation parameters. A summary of the registers is presented in Table 9-1. Read only registers are highlighted in red fill.

Table 9-1 AEC/LEC Control Register Map

AEC		LEC	
Address	Name	Address	Name
0x0300	AEC_CONFIG	0x0380	LEC_CONFIG
0x0301	AEC_RESET	0x0381	LEC_RESET
0x0302	AEC_EC_BELTA	0x0382	LEC_EC_BELTA
0x0303	AEC_AS_COEFF	0x0383	LEC_AS_COEFF
0x0304	Reserved	0x0384	Reserved
Double Talk Detector			
0x0305	AEC_DT_LONG_TC	0x0385	LEC_DT_LONG_TC
0x0306	AEC_DT_SHORT_TC	0x0386	LEC_DT_SHORT_TC
0x0307~ 0x0308	AEC_DT_HANGOVER_TIME	0x0387~ 0x0388	LEC_DT_HANGOVER_TIME
0x0309~ 0x030A	AEC_DT_DV_THRESH	0x0389~ 0x038A	LEC_DT_DV_THRESH
0x030B~ 0x030C	AEC_DT_LONG_THRESH	0x038B~ 0x038C	LEC_DT_LONG_THRESH
0x030D~ 0x030E	AEC_DT_SHORT_THRESH	0x038D~ 0x038E	LEC_DT_SHORT_THRESH
0x030F	AEC_DIVERGENCE	0x038F	LEC_DIVERGENCE
Voice Detector			
0x0310	AEC_VD_LONG_TC	0x0390	LEC_VD_LONG_TC
0x0311	AEC_VD_SHORT_TC	0x0391	LEC_VD_SHORT_TC
0x0312~ 0x0313	AEC_VD_HANGOVER_TIME	0x0392~ 0x0393	LEC_VD_HANGOVER_TIME
0x0314~ 0x0315	AEC_VD_DEV_THRESHOLD	0x0394~ 0x0395	LEC_VD_DEV_THRESHOLD
0x0316~ 0x0317	AEC_VD_LONG_THRESH	0x0396~ 0x0397	LEC_VD_LONG_THRESH

AEC		LEC	
Address	Name	Address	Name
0x0318~ 0x0319	AEC_VD_SHORT_THRESH	0x0398~ 0x0399	LEC_VD_SHORT_THRESH
0x031A~ 0x031B	AEC_CUT_OFF_PWR	0x039A~ 0x039B	LEC_CUT_OFF_PWR
0x31C	Reserved	0x39C	Reserved
0x031D	AEC_VD_AVE_TC	0x039D	LEC_VD_AVE_TC
0x031E~ 0x031F	AEC_VD_AVE_THRESH	0x039E~ 0x039F	LEC_VD_AVE_THRESH
Acoustic Suppressor		**Line Suppressor**	
0x0320	AEC_AS1_BUILD_UP_TIME	0x03A0	LEC_AS1_BUILD_UP_TIME
0x0321- 0x0322	AEC_AS1_MAX_ATTEN	0x03A1~ 0x03A2	LEC_AS1_MAX_ATTEN
0x0323	AEC_AS2_BUILD_UP_TIME	0x03A3	LEC_AS2_BUILD_UP_TIME
0x0324~ 0x0325	AEC_AS2_MAX_ATTEN	0x03A4~ 0x03A5	LEC_AS2_MAX_ATTEN
0x0326~ 0x0327	Reserved	0x03A6~ 0x03A7	Reserved
Noise Suppressor			
0x0328	AEC_NS_POWER_ATTACK_TC	0x03A8	LEC_NS_POWER_ATTACK_TC
0x0329	AEC_NS_ATTEN_TC	0x03A9	LEC_NS_ATTEN_TC
0x032A~ 0x032B	AEC_NS_ACTIVE_THRESHOLD	0x03AA~ 0x03AB	LEC_NS_ACTIVE_THRESHOLD
0x032C~ 0x032D	Reserved	0x03AC~ 0x03AD	Reserved
		Soft Clipping on DT Side	
0x0330	Reserved	0x03B0	LEC_DT_SC_CTRL
0x0331	Reserved	0x03B1	LEC_DT_SC_NORMAL_INDEX
0x0332	Reserved	0x03B2	LEC_DT_SC_LOW_INDEX
0x0333~ 0x0334	Reserved	0x03B3~ 0x03B4	LEC_DT_SC_THRESH

AEC		LEC	
Address	Name	Address	Name
0x0335	Reserved	0x03B5	LEC_DT_SC_POWER_ATTACK_TC
0x0336	Reserved	0x03B6	LEC_DT_SC_GAIN_TC
0x0337	Reserved	0x03B7	Reserved
Soft Clipping on VD Side			
0x0338	AEC_VD_SC_CTRL	0x03B8	LEC_VD_SC_CTRL
0x0339	AEC_VD_SC_NORMAL_INDEX	0x03B9	LEC_VD_SC_NORMAL_INDEX
0x033A	AEC_VD_SC_LOW_INDEX	0x03BA	LEC_VD_SC_LOW_INDEX
0x033B~ 0x033C	AEC_VD_SC_THRESH	0x03BB~ 0x03BC	LEC_VD_SC_THRESH
0x033D	AEC_VD_SC_POWER_ATTACK_TC	0x03BD	LEC_VD_SC_POWER_ATTACK_TC
0x033E	AEC_VD_SC_GAIN_TC	0x03BE	LEC_VD_SC_GAIN_TC
0x033F	Reserved	0x03BF	Reserved
DT Side Power Monitor			
0x0340~ 0x0341	AEC_DT_SHORT_TERM_POWER	0x03C0~ 0x03C1	LEC_DT_SHORT_TERM_POWER
0x0342~ 0x0343	AEC_DT_LONG_TERM_POWER	0x03C2~ 0x03C3	LEC_DT_LONG_TERM_POWER
0x0344~ 0x0345	AEC_DT_POWER_DEVIATION	0x03C4~ 0x03C5	LEC_DT_POWER_DEVIATION
0x0346	AEC_DT_ACTIVE	0x03C6	LEC_DT_ACTIVE
0x0347	Reserved	0x03C7	Reserved
VD Side Power Monitor			
0x0348~ 0x0349	AEC_VD_SHORT_TERM_POWER	0x03C8~ 0x03C9	LEC_VD_SHORT_TERM_POWER
0x034A~ 0x034B	AEC_VD_LONG_TERM_POWER	0x03CA~ 0x03CB	LEC_VD_LONG_TERM_POWER
0x034C~ 0x034D	AEC_VD_POWER_DEVIATION	0x03CC~ 0x03CD	LEC_VD_POWER_DEVIATION
0x034E	AEC_VD_ACTIVE	0x03CE	LEC_VD_ACTIVE
0x034F	Reserved	0x03CF	Reserved
Signal Monitor			
0x0350~ 0x0351	AEC_LRIN	0x03D0~ 0x03D1	LEC_ LRIN

AEC		LEC	
Address	Name	Address	Name
0x0352~ 0x0353	AEC_EOUT	0x03D2~ 0x03D3	LEC_ EOUT
0x0354~ 0x0355	AEC_ASOUT	0x03D4~ 0x03D5	LEC_ ASOUT
0x0356~ 0x0357	AEC_ARIN	0x03D6~ 0x03D7	LEC_ ARIN
0x0358~ 0x0359	AEC_LSOUT	0x03D8~ 0x03D9	LEC_LSOUT
0x035B			
0x035C~ 0x035D			
0x035E~ 0x035F			
AGC Function			
0x0360	AGC_CTRL	0x03E0	Reserved
0x0361	AGC_INIT_GAIN	0x03E1	Reserved
0x0362	AGC_GAIN_HOLD	0x03E2	Reserved
0x0363	AGC_INC_DEC	0x03E3	Reserved
0x0364	AGC_ATK_DCY	0x03E4	Reserved
0x0365	AGC_GAIN_READ	0x03E5	Reserved
0x0366	AGC_STATE	0x03E6	Reserved
0x0367	AGC_POWER_TC	0x03E7	Reserved
0x0368	AGC_PK_TC	0x03E8	Reserved
0x0369~ 0x036A	AGC_PK	0x03E9~ 0x03EA	Reserved
0x036B~ 0x036F	Reserved	0x03EB~ 0x03EF	Reserved
0x0370~ 0x0371	AGC_TARG_CLIP	0x03F0~ 0x03F1	Reserved
0x0372~ 0x0373	AGC_TARG_HI	0x03F2~ 0x03F3	Reserved
0x0374~ 0x0375	AGC_TARG_LO	0x03F4~ 0x03F5	Reserved

AEC		LEC	
Address	Name	Address	Name
0x0376~ 0x0377	AGC_TARG_RB	0x03F6~ 0x03F7	Reserved
0x0378	AGC_NOISE_BIAS	0x03F8	Reserved
0x0379~ 0x037A	AGC_NOISE_HI	0x03F9~ 0x03FA	Reserved
0x037B~ 0x037C	AGC_NOISE_LO	0x03FB~ 0x03FC	Reserved
0x037D~ 0x037F	AGC_NOISE_RB	0x037D~ 0x037F	Reserved

9.2.1 Threshold and Power Calculation

Many registers in this section are used for setting signal threshold power. Threshold powers are measured in dBFS scale (dB relative to a full scale signal) hence a full scale signal will have a power value of 0dBFS, and all signal levels are referenced to this value. A signal that is half the power of the full scale signal will have a power value of 20xlog(1/2) = -6.02dBFS.

To convert register value to actual dBFS value, follow the scheme below:
If register is 1 byte (0x1B), actual power = 20log(0x1B / 0x7F) dBFS
If register is 2 bytes (0x1BAD), actual power = 20log(0x1BAD / 0x7FFF) dBFS
If register is 3 bytes (0x1BADBE), actual power = 20log(0x1BAD BE/ 0x7FFFFF) dBFS
If register is 4 bytes (0x1BADBEEF), actual power = 20log(0x1BADBEEF / 0x7FFFFFFF) dBFS

To convert power (in dBFS) into register value, do the following:
If register is 1 byte, Register value = [10 ^ (power / 20)] * 0x7F
If register is 2 bytes, Register value = [10 ^ (power / 20)] * 0x7FFF
If register is 3 bytes, Register value = [10 ^ (power / 20)] * 0x7FFFFF
If register is 4 bytes, Register value = [10 ^ (power / 20)] * 0x7FFFFFFF

For example, in a 2 bytes register, 0x7FFF corresponds to 0dBFS, 0x3FFF corresponds to -6.02dBFS.

9.3 Control Registers

9.3.1 CONFIG

Address	Access Mode	Value At Reset	Nominal Value
0x300 (AEC) 0x380 (LEC)	R/W	0x96	

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
-	Reserved	DT_NO_ECHO	CUT_THRES_DIR	NO_UPDATE	EC_EN	AS1_EN	AS_PRIORITY

AEC_CONFIG and LEC_CONFIG are general purpose configuration registers for the echo cancellation unit. Various major functions may be enabled or disabled, depending on the state of each flag. Note that echo cancellation may be disabled, yet the updating process for the filter coefficients may remain enabled.

AS_PRIORITY	In half duplex and relative mode, when both VD and DT paths are active, determines which signal path has priority. =1, VD_Active has first priority and AS1 will be active. =0, DT_Active has first priority and AS2 will be active.
AS1_EN	=1, the acoustic suppression 1 (AS1) function will be enabled. =0, disable Note: Acoustic suppressor AS2 only active in half duplex mode (EC_BELTA[5]=1)
EC_EN	Enables adaptive filter echo cancellation. =1, enable =0, disable
NO_UPDATE	=1, the echo cancellation filter parameters will not be updated =0, echo cancellation parameters continuously adapt.
CUT_THRES_DIR	Cut-off threshold direction: =1, the cut off power is applied on the DT side. =0, the cut off power is applied on the VD side. Cut off power is used for power estimation algorithm, defined at reg0x31A and reg0x38A
DT_NO_ECHO	=1, double talk power estimation is based on output power after echo cancellation =0, double talk power estimation detection based on power before echo cancellation

9.3.2 RESET

Address	Access Mode	Value At Reset	Nominal Value
0x301 (AEC) 0x381 (LEC)	R/W	0xE8 / 0xA8	

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
301	BYPASS	SPEEDUP		ALEC RESET	POWER DOWN	RES_COEFF MEM	RES_POW MEM	RES_AIR MEM
381	BYPASS	RESET_FINISH	LEC_8MS	Reserved	POWER DOWN	RES_COEFF MEM	RES_POW MEM	RES_AIR MEM

RES_AIR MEM	Resets the voice sample memory used in Echo Canceller.
RES_POW MEM	Reset the power (channel model error history) memory used in Echo Canceller. Flushing the power and voice sample memories will remove the

Address	Access Mode	Value At Reset	Nominal Value

conversation history.

RES_COEFF MEM — Flushing the FIR coefficient (channel model) memory serves to remove the conversation history and will prompt the filter coefficients in the Echo Canceller to be recalculated from scratch.

POWER DOWN — Halt the AEC / LEC clock with “1” programmed, AEC is default at power down state. Speech signal will bypass AEC / LEC module.

ALEC RESET — While programmed with “1”, AEC / LEC will reset all internal registers.

SPEEDUP — Sets AEC coefficient update speed.
= 00, sets AEC tail length to 32ms with normal update speed.
= 01, sets AEC tail length to 64ms with normal update speed.
= 10, sets AEC tail length to 32ms with fast update speed.
= 11, sets AEC tail length to 32ms and automatically adjust update speed.

BYPASS — When programmed with “1”, speech signal will bypass AEC / LEC module.

LEC_8MS — = 1, sets LEC tail length to 8ms.
= 0, sets LEC tail length to 4ms.

RESET_FINISH — Read only. Will go high after completion of RESET AIR MEM, RESET COEFF MEM, or RESET POWER MEM.

9.3.3 EC_BELTA

Address	Access Mode	Value At Reset	Nominal Value
0x302(AEC) 0x382(LEC)	R/W	0x03	

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
NS_ENABLE	DT_VD_IDLE	HALF DUPLEX	ABSOLUTE	EC_BELTA			

EC_BELTA — EC_BELTA is the echo cancellation update gain and is used to control the rate of change of the echo cancellation filter coefficients. Lowering the value of this configurable constant will increase the inertial delay present in the channel modeling logic. Raising the value of this constant will reduce this inertial delay. Values significantly different from the nominal value for this field will cause the channel modeling algorithm to exhibit instability. In such a situation any echo present in the channel will not be removed.

ABSOLUTE — =1, Absolute mode: the Double Talk detection algorithm is based on absolute value of power on the DT side.
=0, Relative mode: the Double Talk detection algorithm is based on difference of signal power between the DT side and the VD side depending on the setting of DT_COMPARE_DB register in 0x303.

HALF_DUPLEX — =1, EC will enter half-duplex mode and acoustic suppressor2 (AS2) will be enabled.
=0, EC will enter full-duplex mode and acoustic suppressor2 (AS2) disabled.

DT_VD_IDLE — =1, AS2 will be active when no energy is detected in DT and VD channels.

Address	Access Mode	Value At Reset	Nominal Value

PS: This function is only applicable to half-duplex mode.

NS_ENABLE =1, the noise suppressor is enabled
=0, bypass the noise suppressor

9.3.4 AS_COEFF

Address	Access Mode	Value At Reset	Nominal Value
0x303(AEC) 0x383(LEC)	R/W	0x14, 0x04	

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
303	SH_PWR_SEL	AVE_PWR_SE	SH_PWR_ENB	DT_BUF_ENB	DT_COMPARE_DB			
383	SH_PWR_SEL	AVE_PWR_SE	SH_PWR_ENB	BOTH_MODE	DT_COMPARE_DB			

SH_PWR_SEL Selects Air power calculation for echo cancellation algorithm
1: Echo cancellation uses Short Term Air Power to calculate input power.
0: Echo cancellation uses Average Air Power to calculate input power.

AVE_PWR_SE 1: Average Air Power calculation uses zero extension.
0: Average Air Power calculation uses sign extension.

SH_PWR_ENB 1: Disables exponential calculation for Short Term Air Power.
0: Default exponential calculation for Short Term Air Power.

DT_BUF_ENB 1: AEC DT relative mode compares Line power with previous 32 samples from VD. LEC DT relative mode compares Line power with previous 8 samples from VD.
0: DT relative mode compares Line power with 1 sample from VD.

DT_COMPARE_DB In relative mode, if (line power) > (air power x RATIO) then DT is active.
0000: RATIO = 0.000
0001: RATIO = 0.125 (-18dB)
0010: RATIO = 0.250 (-12dB)
0011: RATIO = 0.375 (-8.5dB)
0100: RATIO = 0.500 (-6dB)
..
1111: RATIO = 1.875 (5.46dB)

BOTH_MODE 1: AEC and LEC DT detector only active when BOTH absolute mode and relative mode detection conditions are met.
0: AEC and LEC DT detector uses either absolute mode or relative mode algorithm. (Reg 0x302, 382)

9.4 Double Talk Detector Control Registers

9.4.1 Function

Double talk detector is used to decide whether the signal from the near end is an active voice signal by estimating its signal power. Signal is deemed active if the short term acoustic power exceeds a predetermined threshold or if the short term acoustic power exhibits sudden variations under “absolute” mode, or the relative signal level compared with the signal from far end site in non-absolute mode.

The echo cancellation algorithm will stop adapting to the acoustic characteristics when voice is detected on the near end. If speech is being carried over both near end and far end then the double talk condition occurs. In this situation the acoustic interface signal can exhibit echo-like characteristics and acoustic modeling must be halted.

9.4.2 DT_LONG_TC

Address	Access Mode	Value At Reset	Nominal Value
0x305(AEC) 0x385(LEC)	R/W	0x09	

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Reserved	Reserved	Reserved	Reserved	DT_LONG_TC			

DT_LONG_TC	The DT (Double Talk) long term power estimation’s attacking time constant. This field defines the inertial delay utilized for the long term power estimation at DT side. Raising the value of this field reduces the inertia and will make the estimation more responsive while lowering the field will cause the power estimation algorithm to be less responsive to bursts of energy on the acoustic side.

9.4.3 DT_SHORT_TC

Address	Access Mode	Value At Reset	Nominal Value
0x306(AEC) 0x386(LEC)	R/W	0xBB	

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DT_SHORT_TC_H				DT_SHORT_TC_L			

DT_SHORT_TC	The DT (Double Talk) short term power estimation’s attacking time constant. This field defines the inertial delay utilized for the short term power estimation at DT side. Raising the value of this field reduces the inertia and will make the estimation more responsive while lowering the field will cause the power estimation algorithm to be less responsive to bursts of energy on the acoustic side.
DT_SHORT_TC_H	When input signal power is higher than the estimated short term power, DT_SHORT_TC_H is used for the subsequent power estimation.
DT_SHORT_TC_L	When input signal power is lower or equal to the estimated short term power, DT_SHORT_TC_L is used for the subsequent power estimation.

9.4.4 Double Talk Detector Parameters

Address	Access Mode	Value At Reset	Nominal Value
0x307-0x30E(AEC) 0x387-0x38E(LEC)	R/W	See Below	

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
7	DT_HANGOVER_TIME[15:8]							
8	DT_HANGOVER_TIME[7:0]							
9	DT_DV_THRES[15:8]							
A	DT_DV_THRES[7:0]							
B	DT_LONG_THRESH[15:8]							
C	DT_LONG_THRESH[7:0]							
D	DT_SHORT_THRESH[15:8]							
E	DT_SHORT_THRESH[7:0]							

DT_HANGOVER_TIME (Default 0x0020)	This field defines the inertial delay of the double talk detection algorithm. This delay (in units of 125us) is applied after the loss of double talk detection. If double talk does not reappear during this time then the echo cancellation unit will revert back to acoustic training mode.
DT_DV_THRESH * (Default 0x1998)	Defines the deviation power threshold of double talk detector. This value is used to determine the DT active/inactive condition.
DT_LONG_THRESH * (Default 0x0000)	Minimum power level that constitutes speech over the DT side, as measured by the long term power estimation algorithm.
DT_SHORT_THRESH * (Default 0x1010)	Minimum power level that constitutes speech over the DT side, as measured by the short term power estimation algorithm.

* See 9.2.1Threshold and Power Calculation

9.4.5 DIVERGENCE

Address	Access Mode	Value At Reset	Nominal Value
0x30F(AEC) 0x38F(LEC)	R/W	0x0F	

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DIVERGENCE	Reserved	DIVERGENCE THRESHOLD					

DIVERGENCE THRESHOLD	When the signal power ratio Eout/Lrin > DIVERGENCE_THRESHOLD (see Figure 9-1), a divergence condition is detected. When a divergence condition is detected, echo cancellation coefficients are cleared and adaptation starts over again. DIVERGENCE_THRESHOLD is an unsigned hexadecimal value, hence 0x3F is the maximum value (representing decimal 63) and 0x00 is the minimum value (representing decimal 0)
DIVERGENCE	Once a divergence condition is detected, this bit stays high, until a write of any value occurs on this register.

9.5 Voice Detector Control Registers

9.5.1 Function

The acoustic modeling algorithm requires a measure of the instantaneous speech power on the Far End of the echo canceller (this is termed the VD Voice Detect side). To detect the double talk condition an estimate of the long term average power of the VD side is also required.

9.5.2 VD_LONG_TC

Address	Access Mode	Value At Reset	Nominal Value
0x310(AEC) 0x390(LEC)	R/W	0x09	

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Reserved	Reserved	Reserved	Reserved	VD_LONG_TC			

VD_LONG_TC	VD long term average power estimation's attacking time constant. This field defines the inertial delay utilized for the long term power estimation of VD. Raising the value of this field reduces the inertia and will make the estimation more responsive whilst lowering the field will cause the power estimation algorithm to be less responsive to bursts of energy on the VD side.

9.5.3 VD_SHORT_TC

Address	Access Mode	Value At Reset	Nominal Value
0x311(AEC) 0x391(LEC)	R/W	0xBB	

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
VD_SHORT_TC_H				VD_SHORT_TC_L			

VD_SHORT_TC	VD short term power average estimation attacking time constant. This field defines the inertial delay utilized for the short term power estimation of VD. Raising the value of this field reduces the inertia and will make the estimation more responsive whilst lowering the field will cause the power estimation algorithm to be less responsive to bursts of energy on the VD side.
VD_SHORT_TC_H	When input signal power is higher than the estimated short term power, VD_SHORT_TC_H is used for the subsequent power estimation.
VD_SHORT_TC_L	When input signal power is lower or equal to the estimated short term power, VD_SHORT_TC_L is used for the subsequent power estimation.

9.5.4 Voice Detector Parameters

Address	Access Mode	Value At Reset	Nominal Value
0x312-0x31B(AEC) 0x392-0x39B(LEC)	R/W	See Below	

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
VD_HANGOVER_TIME[15:8]							

2

	Address	Access Mode	Value At Reset	Nominal Value
3	VD_HANGOVER_TIME[7:0]			
4	VD_DEV_THRESHOLD[15:8]			
5	VD_DEV_THRESHOLD[7:0]			
6	VD_LONG_THRESH[15:8]			
7	VD_LONG_THRESH[7:0]			
8	VD_SHORT_THRESH[15:8]			
9	VD_SHORT_THRESH[7:0]			
A	CUT_OFF_POWER[15:8]			
B	CUT_OFF_POWER[7:0]			

VD_HANGOVER_TIME (Default 0x0009)	This field defines the inertial delay of the voice detection algorithm for VD side. Following the detection of the speech on the VD side there is a programmable inertial delay (in units of 125us) following the disappearance of the speech signal. For the duration of this delay period the speech is assumed to remain. If speech does not reappear during this window then the echo cancellation unit will revert back to acoustic training mode.
VD_DEV_THRESHOLD * (Default 0x1998)	Defines the deviation power threshold of VD. This value is used to determine VD active/inactive condition.
VD_LONG_THRESH * (Default 0x1998)	Defines the minimum power level that constitutes speech over the VD side, as measured by the long term power estimation algorithm.
VD_SHORT_THRESH * (Default 0x1038)	Minimum power level that constitutes speech over the VD side, as measured by the short term power estimation algorithm.
CUT_OFF_POWER * (Default 0x1998)	Configurable bias registers for power estimation. This field defines the zero reference for the power estimation algorithm.

* See 9.2.1Threshold and Power Calculation

VD_AVE_TC

Address	Access Mode	Value At Reset	Nominal Value
0x31D(AEC) 0x39D(LEC)	R/W	0x0B	

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Reserved	Reserved	Reserved	Reserved	AVE_SHORT_TC			

AVE_SHORT_TC	Average short term power estimation attacking time constant. This field defines the inertial delay utilized for the average short term power estimation. Raising the value of this field reduces the inertia and will make the estimation more responsive whilst lowering the field will cause the power estimation algorithm to be less responsive to bursts of energy.

9.5.5 VD_AVE_THRESH

Address	Access Mode	Value At Reset	Nominal Value
0x31E-0x31F(AEC) 0x39E-0x39F(LEC)	R/W	See Below	

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
E	VD_AVE_THRESH[15:8]							
F	VD_AVE_THRESH[7:0]							

VD_AVE_THRESH * (Default 0x0000) — Minimum average power level that constitutes speech over the VD side, as measured by the short term power estimation algorithm.

* See 9.2.1Threshold and Power Calculation

9.6 AS1 & AS2 Control Registers

9.6.1 Function

The acoustic suppression (AS1& AS2) units will insert configurable attenuation factors into the voice path. The attenuation will switch between a maximum and minimum value depending on the presence or absence of speech on the voice path. When speech is present the attenuation will converge towards the minimum value. When speech is absent the attenuation will converge towards the maximum value. Whether speech is present is decided by the Double Talk Detector (DT) and the Voice Detector (VD). The attenuation factor will not switch abruptly between these two factors but will exponentially converge from one to the other. The suppression units are used in half-duplex mode.

9.6.2 AS1_BUILD_UP_TIME

Address	Access Mode	Value At Reset	Nominal Value
0x320(AEC) 0x3A0(LEC)	R/W	0x77	

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
AS1_BUILD_UP_TIME_POS				AS1_BUILD_UP_TIME_NEG			

Time constant determining how fast AS1 gain ramps from max to min gain (AS1_BUILD_UP_TIME_POS) and from min to max gain (AS1_BUILD_UP_TIME_NEG). Smaller time constant represents faster change in gain.

9.6.3 AS1_MAX_ATTEN

Address	Access Mode	Value At Reset	Nominal Value
0x321-0x322(AEC) 0x3A1-0x3A2(LEC)	R/W	0x1CA8	

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
1	AS1_MAX_ATTEN[15:8]							
2	AS1_MAX_ATTEN[7:0]							

Maximum attenuation value will be utilized by the acoustic suppression algorithm. Actual maximum attenuation = 20 x log (AS1_MAX_ATTEN / 0xFFFF) (dB). The maximum value of this field (0xFFFF) provides an attenuation factor of 1 (0dB – no attenuation). The minimum (0x0000) value provides an attenuation factor of 0.

9.6.4 AS2_BUILD_UP_TIME

Address	Access Mode	Value At Reset	Nominal Value
0x323(AEC) 0x3A3(LEC)	R/W	0x77	

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
AS2_BUILD_UP_TIME_POS				AS2_BUILD_UP_TIME_NEG			

Time constant determining how fast AS2 gain (towards target) ramps from max to min gain (AS2_BUILD_UP_TIME_POS) and from min to max gain (AS2_BUILD_UP_TIME_NEG). Smaller time constant represents faster change in gain.

9.6.5 AS2_MAX_ATTEN

Address	Access Mode	Value At Reset	Nominal Value
0x324-0x325(AEC) 0x3A4-0x3A5(LEC)	R/W	0x1CA8	

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
1	AS1_MAX_ATTEN[15:8]							
2	AS1_MAX_ATTEN[7:0]							

Maximum attenuation value will be utilized by the acoustic suppression algorithm. Actual maximum attenuation = 20 x log (AS1_MAX_ATTEN / 0xFFFF) (dB). The maximum value of this field (0xFFFF) provides an attenuation factor of 1 (0dB – no attenuation). The minimum (0x0000) value provides an attenuation factor of 0.

9.7 Noise Suppressor Registers

9.7.1 Function

Noise suppressor is used to attenuate noise from microphone/line to reduce residue echo from the near end. If power of near-end noise (P_{NS}) < NS_ACTIVE_THRESHOLD, then Noise Suppressor will attenuate near-end signal by (NS_INDEX+1) dB. The parameters NS_FALL_TC and NS_RISE_TC control gain attack and release speeds.

9.7.2 NS_POWER_ATTACK_TC

Address	Access Mode	Value At Reset	Nominal Value
0x328(AEC) 0x3A8(LEC)	R/W	0xBB	

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
NS_POWER_ATTACK_TC[3:0]				NS_INDEX[3:0]			

NS_INDEX — Defines the gain of the noise suppressor Gain = -(NS_INDEX+1)dB.

NS_POWER_ATTACK_TC — This field defines the time constant of the power estimation of signal that enters the nose suppressor module. Noise suppression is active if noise power is less than noise threshold. Raising the value of this field reduces the inertia and will make the estimation more responsive whilst lowering the field will cause the power estimation algorithm to be less responsive to bursts of energy on the input of the noise suppressor.

9.7.3 NS_ATTEN _TC

Address	Access Mode	Value At Reset	Nominal Value
0x329(AEC) 0x3A9(LEC)	R/W	0xBB	

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
NS_FALL_TC [3:0]				NS_RISE_TC [3:0]			

NS_RISE_TC[3:0] — Define the time constant of noise suppressor gain transition from the gain specified by NS_INDEX to 0dB. Larger value results in faster transition.

NS_FALL_TC[3:0] — Define the time constant of Noise suppressor gain transition from 0dB to the gain specified by NS_INDEX. Larger value results in faster transition.

9.7.4 NS_ACTIVE_THRESHOLD

Address	Access Mode	Value At Reset	Nominal Value
0x32A-0x32B(AEC) 0x3AA-0x3AB(LEC)	R/W	0x03E8	

Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
A	NS_ACTIVE_THRESHOLD[15:8]						
B	NS_ACTIVE_THRESHOLD[7:0]						

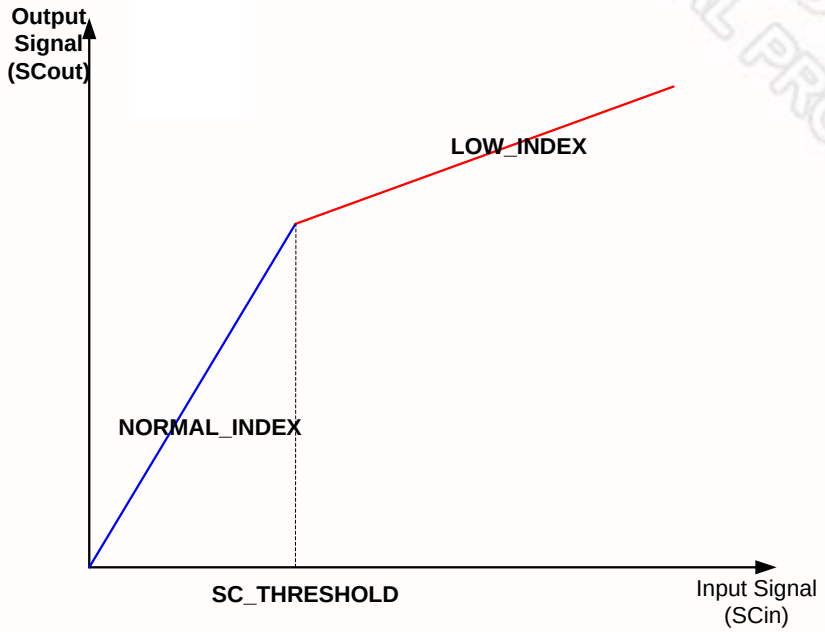
When signal energy is less than NS_ACTIVE_THRESHOLD the noise suppressor function will activate. See 9.2.1Threshold and Power Calculation

9.8

9.9 Soft Clip (SC) Control Registers

9.9.1 Functional Description

The echo cancellation cannot operate effectively on non-linear clipped signals. To reduce non-linearity due to signal clipping, a soft clipping function is available. Soft clipping control is available in both signal path directions on the LEC and on the output path of the AEC. Soft Clip gain is control by SC_NORMAL_INDEX and SC_LOW_INDEX. Normal Soft Clip gain is SC_NORMAL_INDEX. If input signal level is greater than SC_THRESHOLD the SC gain is changed to SC_LOW_INDEX for reducing clipping distortion.



9.9.2 SC_CTRL

Address	Access Mode	Value At Reset	Nominal Value
0x338 (AEC VD) 0x3B0(LEC DT) 0x3B8(LEC VD)	R/W	0x00	

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
338, 3B8	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	ADAPTIVE_THRES	SC_EN
3B0	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	SC_EN

SC_EN =1, Enable the soft clipping function.
=0, Disable the soft clipping function.

Address	Access Mode	Value At Reset	Nominal Value

ADAPTIVE_THRES =1, If DT long term line power > DT short term threshold, DT short term threshold automatically increases to same level as DT long term line power.

9.9.3 SC_NORMAL_INDEX

Address	Access Mode	Value At Reset	Nominal Value
0x339 (AEC VD) 0x3B1(LEC DT) 0x3B9(LEC VD)	R/W	0x00	

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Reserved	GAIN_INDEX[6:0]						

GAIN_INDEX[6:0] This value sets the gain in the normal gain region of the soft clip transfer function. Gain selection range is shown in Table 9-2.

Table 9-2 SC Gain Table

GAIN_INDEX[6:0] HEX Value	Gain	GAIN_INDEX[6:0] HEX Value	Gain
0x00	0 dB (default)	0x7F	- 0.5 dB
0x01	0.5 dB	0x7E	- 1.0 dB
0x02	1.0 dB	0x7D	- 1.5 dB
0x03	1.5 dB	0x7C	- 2.0 dB
0x04	2.0 dB	0x7B	- 2.5 dB
0x05	2.5 dB	0x7A	- 3.0 dB
0x06	3.0 dB	0x79	- 3.5 dB
0x07	3.5 dB	0x78	- 4.0 dB
0x08	4.0 dB	0x77	- 4.5 dB
0x09	4.5 dB	0x76	- 5.0 dB
0x0A	5.0 dB	0x75	- 5.5 dB
0x0B	5.5 dB	0x74	- 6.0 dB
0x0C	6.0 dB	0x73	- 6.5 dB
0x0D	6.5 dB	0x72	- 7.0 dB
0x0E	7.0 dB	0x71	- 7.5 dB
0x0F	7.5 dB	0x70	- 8.0 dB
0x10	8.0 dB	0x6F	- 8.5 dB
0x11	8.5 dB	0x6E	- 9.0 dB
0x12	9.0 dB	0x6D	- 9.5 dB
0x13	9.5 dB	0x6C	- 10.0 dB
0x14	10.0 dB	0x6B	- 10.5 dB
0x15	10.5 dB	0x6A	- 11.0 dB
0x16	11.0 dB	0x69	- 11.5 dB

0x17	11.5 dB	0x68	- 12.0 dB
0x18	12.0 dB	0x40	Mute

9.9.4 SC_LOW_INDEX

Address	Access Mode	Value At Reset	Nominal Value
0x33A (AEC VD) 0x3B2(LEC DT) 0x3BA(LEC VD)	R/W	0x00	

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Reserved	GAIN_INDEX[6:0]						

GAIN_INDEX[6:0] This value sets the gain in the low gain region of the soft clip transfer function. Gain selection range is shown in Table 9-2.

9.9.5 SC_THRESH

Address	Access Mode	Value At Reset	Nominal Value
0x33B-0x33C(AEC VD) 0x3B3-0x3B4(LEC DT) 0x3BB-0x3BC(LEC DT)	R/W	0x1000	

Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
SC_THRESHOLD[15:8]							
SC_THRESHOLD[7:0]							

SC_THRESHOLD is used to determine the selection of Soft Clip gain. When the input power is larger than SC_THRESHOLD, SC_LOW_INDEX is used to set gain, otherwise SC_NORMAL_INDEX is used. See 9.2.1Threshold and Power Calculation

9.9.6 SC_POWER_ATTACK_TC

Address	Access Mode	Value At Reset	Nominal Value
0x33D(AEC VD) 0x3B5(LEC DT) 0x3BD(LEC VD)	R/W	0x07	

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Reserved	Reserved	Reserved	Reserved	TC[3:0]			

This time constant is used to calculate the short term power estimation for soft clipping threshold comparison. Raising the value of this field reduces the inertia and will make the estimation more responsive whilst lowering the field will cause the power estimation algorithm to be less responsive to bursts of energy.

9.9.7 SC_GAIN_TC

Address	Access Mode	Value At Reset	Nominal Value
0x33E(AEC VD) 0x3B6(LEC DT) 0x3BE(LEC VD)	R/W	0x07	

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Reserved	Reserved	Reserved	Reserved	TC[3:0]			

When the soft clip gain is switched between normal and low gains, an embedded smoothing function is applied to the gain transition. This time constant controls the smoothing. Larger time constant results in slower transition.

9.10 State Read back Registers

9.10.1 Functional Description

The following registers allow read back of internal parameter measurements to facilitate configuration of echo cancellation blocks.

9.10.2 Power Monitor

Address	Access Mode	Value At Reset	Nominal Value
0x340-0x34E(AEC) 0x3C0-0x3CE(LEC)	R	See Below	

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	DT_SHORT_TERM_POWER[15:8]							
1	DT_SHORT_TERM_POWER [7:0]							
2	DT_LONG_TERM_POWER [15:8]							
3	DT_LONG_TERM_POWER [7:0]							
4	DT_POWER_DEVIATION [15:8]							
5	DT_POWER_DEVIATION [7:0]							
6	reserved							DT_ACTIVE
7	reserved							
8	VD_SHORT_TERM_POWER[15:8]							
9	VD_SHORT_TERM_POWER [7:0]							
A	VD_LONG_TERM_POWER [15:8]							
B	VD_LONG_TERM_POWER [7:0]							
C	VD_POWER_DEVIATION [15:8]							

Address	Access Mode	Value At Reset	Nominal Value

D	VD_POWER_DEVIATION [7:0]							
E	reserved							VD_ACTIVE

SHORT_TERM_POWER	Short term power estimate on DT/VD side
LONG_TERM_POWER	Long term power estimate on DT/VD side.
POWER_DEVIATION	Power deviation estimation on DT/VD side.
DT_ACTIVE	1: Double-talk activity
VD_ACTIVE	1: Voice Detection activity.

9.10.3 Signal Monitor

Address	Access Mode	Value At Reset	Nominal Value
0x350-0x359(AEC) 0x3D0-0x3D9(LEC)	R	See Below	

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	LRIN[15:8]							
1	LRIN[7:0]							
2	EOUT[15:8]							
3	EOUT[7:0]							
4	ASOUT[15:8]							
5	ASOUT[7:0]							
6	ARIN[15:8]							
7	ARIN[7:0]							
8	LSOUT[15:8]							
9	LSOUT[7:0]							

LRIN	Signal PCM value at Lrin
EOUT	Signal PCM value at Eout
ASOUT	Signal PCM value at Asout
ARIN	Signal PCM value at Arin
LSOUT	Signal PCM value at Lsout

See Figure 9-1: Signal flow through the Acoustic Echo Canceller (AEC) in the speech Processor.

9.11

9.12 Automatic Gain Control

AGC function will automatically adjust the gain to the target voice level specified by the parameters AGC_TARG_HI and AGC_TARG_LO. The actions of the AGC out of the target range are tuned by the parameters AGC_NOISE_HI, AGC_NOISE_LO, GAIN_MAX, GAIN_MIN and the time constant parameters. The operation of the AGC is shown in the figure below.

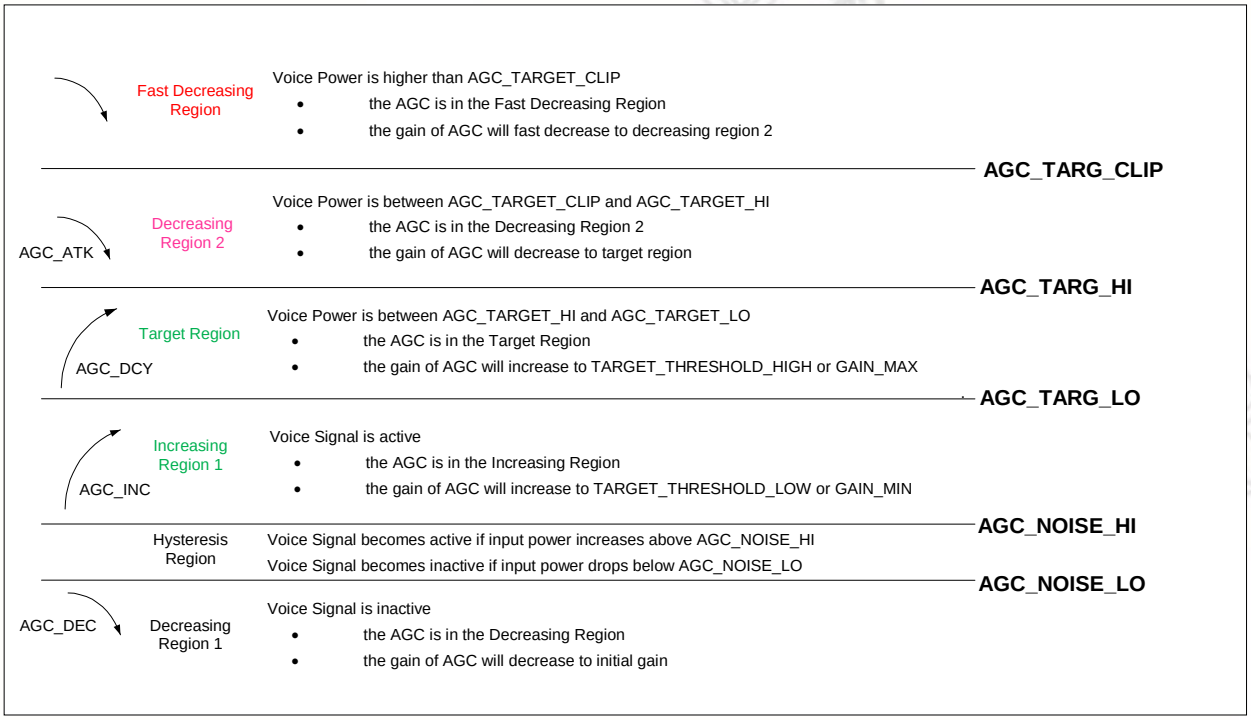


Figure 9-3 AGC Operation regions.

The AGC is activated when the signal short term energy increases above AGC_NOISE_HI. It stays active until the signal short term power drops below AGC_NOISE_LO.

Once activated, the AGC enters the Increasing Region 1. This region increases the gain with a time constant AGC_INC until either signal enters the target region or GAIN_MAX gain is reached. The goal of this region is to provide a quick reaction to voice energy and supply some gain.

Once into the target region, the gain is incremented with a time constant AGC_DCY to a maximum gain of GAIN_MAX, goal of this region is for slower response to keep the signal in target region. If signal amplitude goes above AGC_TARG_HI, the AGC goes into Decreasing Region 2.

In Decreasing Region 2, gain is gradually decreased with a time constant of AGC_ATK to pull signal back into the target region.

If a large signal is applied suddenly, the AGC signal could shoot above AGC_TARG_CLIP. In this case the AGC will enter fast decreasing region to quickly reduce its gain.

If the signal is small enough that it is below the AGC_NOISE_LO, the AGC treats the signal as noise and will not try to gain it up. The AGC will stay in decreasing region 1 and gain will be gradually decremented by with a time constant of AGC_DEC down to the initial gain.

In addition to the above, the gain changing action of the AGC can be controlled by the HOLD time. The hold timer is reset whenever the AGC moves from the Decreasing region to the Target region. If a hold time is selected, no gain changes will occur until after this until target time is reached. This has the effect of keeping the gain constant rather than “hunting” for an ideal gain.

9.12.1 AGC_CTRL

Address	Access Mode	Value At Reset	Nominal Value
0x0360	R/W	0x00	

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
AGC_EN	AGC_ONLY	GAIN_MIN			GAIN_MAX		

AGC_EN	Set 1: Enable AGC. Set 0: Disable AGC
AGC_ONLY	Set 1: The AEC block is bypassed and AGC operates from AEC input.
GAIN_MIN	Refer to Figure 9-3. When the peak of AGC output is smaller than AGC_TARG_LO and greater than AGC_NOISE_HI, the AGC is in Increasing Region 1. The AGC gain will increase up to GAIN_MIN 000: The low gain is $20*\log1.25^2$, which is 3.9dB. 001: The low gain is $20*\log1.50^2$, which is 7.0dB. 010: The low gain is $20*\log1.75^2$, which is 9.7dB. 011: The low gain is $20*\log2.00^2$, which is 12.0dB. 100: The low gain is $20*\log2.25^2$, which is 14.1dB. 101: The low gain is $20*\log2.50^2$, which is 15.9dB. 110: The low gain is $20*\log2.75^2$, which is 17.6dB. 111: The low gain is $20*\log3.00^2$, which is 19.1dB.
GAIN_MAX	Refer to Figure 9-3. When the peak of AGC out is smaller than AGC_TARG_HI and greater than AGC_TARG_LO, the AGC is in Target Region. The AGC gain will increase up to GAIN_MAX. 000: The high gain is $20*\log1.25^2$, which is 3.9dB. 001: The high gain is $20*\log1.50^2$, which is 7.0dB. 010: The high gain is $20*\log1.75^2$, which is 9.7dB. 011: The high gain is $20*\log2.00^2$, which is 12.0dB. 100: The high gain is $20*\log2.25^2$, which is 14.1dB. 101: The high gain is $20*\log2.50^2$, which is 15.9dB. 110: The high gain is $20*\log2.75^2$, which is 17.6dB. 111: The high gain is $20*\log3.00^2$, which is 19.1dB.

The purpose of having Increasing Region 1 and Target Region is to have different converge rates for different levels of input signal.

9.12.2 AGC_INIT_GAIN

Address	Access Mode	Value At Reset	Nominal Value
0x0361	R/W	0x88	

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
IG	AGC_INITIAL_GAIN						

AGC_INITIAL_GAIN Determines initial gain when the AGC is enabled. In unsigned 4.3 format. Thus actual initial gain = 20 x log (AGC_INITIAL_GAIN / 0x08) dB.

IG Initial Gain bit: if set to 1 then AGC when disabled resets to AGC_INITIAL_GAIN

9.12.3 AGC_GAIN_HOLD

Address	Access Mode	Value At Reset	Nominal Value
0x0362	R/W	0x00	

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
AGC_HOLD			AGC_HOLD_CNT				

AGC_HOLD The unit of hold time = (2^AGC_HOLD) * 1 ms.

AGC_HOLD_CNT Together with AGC_HOLD determines how long the AGC will hold gain constant after entering target region before increasing gain.
The hold time is AGC_HOLD * AGC_HOLD_CNT.

9.12.4 AGC_INC_DEC

Address	Access Mode	Value At Reset	Nominal Value
0x0363	R/W	0x00	

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Reserved	AGC_INC			Reserved	AGC_DEC		

AGC_INC Determines the time constant at which gain increases when AGC is in Increasing Region 1.
The increase time is (2^AGC_INC) * 4us.

AGC_DEC Determines the time constant at which the gain reduces when AGC is in Decreasing Region 1.
The decrease time is (2^AGC_DEC) * 4us.

9.12.5 AGC_ATK_DCY

Address	Access Mode	Value At Reset	Nominal Value
0x0364	R/W	0x00	

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Reserved	AGC_DCY			Reserved	AGC_ATK		

AGC_DCY Controls the decay time constant: that is how fast the gain increases when AGC is in Target Region.
The DCY time is (2^AGC_DCY) * 16us per gain step.

AGC_ATK Controls the attack time constant: that is how fast the gain decreases when

Address	Access Mode	Value At Reset	Nominal Value

AGC is in Decreasing Region 2.
The ATK time is (2^AGC_ATK) * 16us per gain step.

9.12.6 AGC_GAIN_READ

Address	Access Mode	Value At Reset	Nominal Value
0x0365	R	0x00	

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
AGC_GAIN_READ							

AGC_GAIN_READ Returns the current AGC gain in unsigned 4.4 format. Actual gain = 20 x log (AGC_GAIN_READ / 0x10) dB

9.12.7 AGC_STATE

Address	Access Mode	Value At Reset	Nominal Value
0x0366	R	0x00	

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
VOICE ACTIVE	HOLD_END	LT_MAX	FAST_DEC	INC	DCY	DEC	ATK

VOICE ACTIVE Signal Energy level is higher than AGC_NOISE_H.
HOLD_END Indicates AGC hold time has reached 0.
LT_MAX Indicates AGC gain is less than GAIN_MAX.
INC Gain is increasing. AGC is in Increasing Region 1.
DCY Gain is increasing. AGC is in Target Region.
DEC Gain is decreasing to unit gain. AGC is in Decreasing Region 1.
ATK Gain is decreasing. AGC is in Decreasing Region 2.
FAST_DEC Gain is decreasing. AGC is in Fast Decreasing Region.

9.12.8 AGC_PWR_TC

Address	Access Mode	Value At Reset	Nominal Value
0x0367	R/W	0x00	

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
AGC_ST_PWR_TC_H				AGC_ST_PWR_TC_L			

AGC_ST_PWR_TC Determines the time constant for calculating the short term noise energy used in the AGC block. The short term noise energy is used to determine whether the AGC should be active. Larger time constant makes short term noise energy responds faster to signal change.
AGC_ST_PWR_TC_H When input signal power is higher than the estimated short term energy, AGC_ST_PWR_TC_H is used for the subsequent energy estimation.
AGC_ST_PWR_TC_L When input signal power is lower or equal to the estimated short term energy, AGC_ST_PWR_TC_L is used for the subsequent energy estimation.

9.12.9 AGC_PK_TC

Address	Access Mode	Value At Reset	Nominal Value
0x0368	R/W	0x00	

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
-	Reserved	Reserved	Reserved	Reserved	Reserved	AGC_PEAK_REL	

AGC_PEAK_REL Determines the release time of peak detector. Peak detector is used to detect the peak of AGC output signal and decide which region the AGC should operate in. The REL time is (2^AGC_PEAK_REL) * 4us.

9.12.10AGC_PK

Address	Access Mode	Value At Reset	Nominal Value
0x369-0x36A	R	0x00	

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
69	Reserved	AGC_PK[14:8]						
6A	AGC_PK[7:0]							

AGC_PK Read back for monitoring peak detector value. See 9.2.1Threshold and Power Calculation

9.12.11AGC TARGETS

Address	Access Mode	Value At Reset	Nominal Value
0x0370-0x0377	R/W	0x00	

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
70	AGC_TARG_CLIP[15:8]							
71	AGC_TARG_CLIP [7:0]							
72	AGC_TARG_HI[15:8]							
73	AGC_TARG_HI [7:0]							
74	AGC_TARG_LO[15:8]							
75	AGC_TARG_LO [7:0]							
76	AGC_TARG_RB[15:8]							
77	AGC_TARG_RB [7:0]							

AGC_TARG_CLIP * Sets the very high limit of target region. It is in 0.16 format representing decimal ratio of full scale.

AGC_TARG_HI * Sets the high limit of target region. It is in 0.16 format representing decimal ratio of full scale.

AGC_TARG_LO * Sets the low limit of target region. It is in 0.16 format representing decimal ratio of full scale.

AGC_TARG_RB * Read only for monitoring current AGC output signal level.

* See 9.2.1Threshold and Power Calculation

9.12.12AGC NOISE PARAMETERS

Address	Access Mode	Value At Reset	Nominal Value
0x0378-0x037D	R/W	0x00	

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
78	AGC_NOISE_BIAS[7:0]							
79	AGC_NOISE_HI[15:8]							
7A	AGC_NOISE_HI[7:0]							
7B	AGC_NOISE_LO[15:8]							
7C	AGC_NOISE_LO[7:0]							
7D	AGC_NOISE_RB[23:16]							
7E	AGC_NOISE_RB[15:8]							
7F	AGC_NOISE_RB [7:0]							

AGC_NOISE_THD_BIAS Sets the bias for AGC_NOISE_HI and AGC_NOISE_LO. See below.

AGC_NOISE_HI * Sets the high limit of noise region, biased by AGC_NOISE_THD_BIAS. The actual noise high limit is AGC_NOISE_THD_BIAS[7:0] x 2^{-8} + AGC_NOISE_HI[15:0] x 2^{-24} of the full scale signal.

AGC_NOISE_LO * Sets the low limit of noise region, biased by AGC_NOISE_THD_BIAS The actual noise low limit is AGC_NOISE_THD_BIAS[7:0] x 2^{-8} + AGC_NOISE_LO[15:0] x 2^{-24} of the full scale signal.

AGC_NOISE_RB * Read only for monitoring current noise power estimate. It is in 0.24 format representing decimal ratio of full scale.

* See 9.2.1Threshold and Power Calculation

10. SPI COMMANDS

The ISD61S00 provides SPI commands including: audio play and record commands, device status commands, digital commands, and device configuration commands.

The following section contains a list of all SPI commands and their function.

Table 10-1 SPI Commands

Instructions	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4 … Byte n	Description
PLAY_VP	0xA6	Index[15:8]	Index[7:0]			Play Voice Prompt Index

Instructions	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4 … Byte n	Description
PLAY_VP@Rn	0xAE	N=0…7				Play Voice Prompt; Index @ Rn
PLAY_VP_LP	0xA4	Index[15:8]	Index[7:0]	LoopCnt[15:8]	LoopCnt[7:0]	Loop Play Voice Prompt Index
PLAY_VP_LP@Rn	0xB2	N=0…7	LoopCnt[15:8]	LoopCnt[7:0]		Loop Play Voice Prompt; Index @ Rn
STOP_LP	0x2E					Stop the active PLAY_VP_LP or PLAY_VP_LP@Rn
EXE_VM	0xB0	Index[15:8]	Index[7:0]			Execute voice macro Index
EXE_VM@Rn	0xBC	N=0…7				Execute voice macro Index @ Rn
REC_MSG	0x38					Record message
REC_MSG@	0x3A	A[23:16]	A[15:8]	A[7:0]		Record message starting at address A.
PLAY_MSG@	0x3C	A[23:16]	A[15:8]	A[7:0]	Off[15:8], Off[7:0]	Play message starting at address A offset by OFF sectors.
PLAY_SIL	0xA8	LEN[7:0]				Play silence for LEN*32ms
STOP	0x2A					STOP current playback or record operation.
ERASE_MSG@	0x3E	A[23:16]	A[15:8]	A[7:0]		Erase message starting at address A.
SPI_SND_AUD	0xAA	D0[7:0]	D0[15:8]	D1[7:0]	D1[15:8] …Dn[7:0] Dn[15:8]	Send 16 bit PCM audio data [low-byte, high-byte] to ISD61S00 via SPI interface when doing REC_MGS
SPI_RCV_AUD	0xAC	D0[7:0]	D0[15:8]	D1[7:0]	D1[15:8] …Dn[7:0] Dn[15:8]	Receive 16 bit PCM audio data [low-byte, high-byte] from ISD61S00 via SPI interface when doing PLAY_MSG
SPI_SND_CMPR	0xDC	D0[7:0]	D1[7:0]	D2[7:0]	D3[7:0] …Dn[7:0]	Send compressed audio data to ISD61S00 via SPI interface to direct write to flash while doing REC_MSG/REC_MSG@
SPI_RCV_CMPR	0xBE	D0[7:0]	D0[15:8]	D1[7:0]	D3[7:0] …Dn[7:0]	Direct read compressed audio data in ISD61S00 flash via SPI interface while doing PLAY_MSG@
SPI_SND_DEC	0xC0	D0[7:0]	D1[7:0]	D2[7:0]	D3[7:0] …Dn[7:0]	Send compressed audio data to ISD61S00 via SPI interface for decoding.
SPI_RCV_ENC	0xC2	D0[7:0]	D1[7:0]	D2[7:0]	D3[7:0] …Dn[7:0]	Receive compressed (encoded) audio data from ISD61S00 via SPI interface.

Instructions	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4 … Byte n	Description
READ_STATUS	0x40	XX	XX	XX	XX	Query status of ISD61S00.
READ_INT	0x46	XX	XX	XX	XX	Query status and clear interrupt flags of ISD61S00.
RD_MSG_ADD	0x42	XX	XX	XX	XX, XX	Query message address details of audio record. Returns start address A and current sector length LEN.
RD_MSG_LEN	0x44	XX	XX			Query current sector length LEN of current playback or record operation.
READ_ID	0x48	XX	XX	XX	XX	Read device ID of ISD61S00.
DIG_READ	0xA2	A[23:16]	A[15:8]	A[7:0]	XX, … XX	Read digital data from address A.
DIG_WRITE	0xA0	A[23:16]	A[15:8]	A[7:0],	D0[7:0], … Dn[7:0]	Write digital data from address A.
ERASE_MEM	0x24	SA[23:16]	SA[15:8]	SA[7:0]	EA[23:16], EA[15:8], EA[7:0]	Erase sectors of memory from sector containing SA to sector containing EA.
CHIP_ERASE	0x26	0x01				Initiate a mass erase of memory
CHECKSUM	0xF2	EA[23:16]	EA[15:8]	EA[7:0]		Calculate checksum from very beginning to the specified end address.
PWR_UP	0x10					Power up ISD61S00
PWR_DN	0x12					Power down ISD61S00
WR_CFG_REG	0x80 + REG[9:8]	REG[7:0]	D0[7:0], ...Dn[7:0]			Write data D0...Dn to configuration register(s) starting at configuration register REG.
RD_CFG_REG	0x90 + REG[9:8]	REG[7:0]	XX, ...XX			Read configuration register(s) starting at configuration register REG.

Each command will be accepted if certain conditions are met as in the following table, or a CMD_ERR interrupt will be generated and the command ignored.

Table 10-2 Commands vs. Status

Instructions	Op Code	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
		PD	DBUF_RDY	INT	RM_FUL	-	VM_BSY	CBUF_FUL	CMD_BSY
PLAY_VP	0xA6	0	***1***	x	x	-	0	0	x
PLAY_VP@Rn	0xAE	0	***1***	x	x	-	0	0	x
PLAY_VP_LP	0xA4	0	***1***	x	x	-	0	0	x

PLAY_VP_LP@Rn	0xB2	0	***1***	x	x	-	0	0	x
STOP_LP	0x2E	0	***1***	x	x	-	x	x	x
EXE_VM	0xB0	0	***1***	x	x	-	0	0	0
EXE_VM@Rn	0xBC	0	***1***	x	x	-	0	0	0
REC_MSG	0x38	0	***1***	x	0	-	0	0	x
REC_MSG@	0x3A	0	***1***	x	0	-	0	0	x
PLAY_MSG@	0x3C	0	***1***	x	x	-	0	0	x
PLAY_SIL	0xA8	0	***1***	x	x	-	0	0	x
STOP	0x2A	0	***1***	x	x	-	x	x	x
ERASE_MSG@	0x3E	0	***1***	x	x	-	0	0	0
SPI_SND_AUD	0xAA	0	***1***	x	x	-	x	x	x
SPI_RCV_AUD	0xAC	0	***1***	x	x	-	x	x	x
SPI_SND_CMPR	0xDC	0	***1***	x	x	-	x	x	x
SPI_RCV_CMPR	0xBE	0	***1***	x	x	-	x	x	x
SPI_SND_DEC	0xC0	0	***1***	x	x	-	0	0	0
SPI_RCV_ENC	0xC2	0	***1***	x	x	-	0	0	0
READ_STATUS	0x40	x	x	x	x	-	x	x	x
READ_INT	0x46	x	x	x	x	-	x	x	x
RD_MSG_ADD	0x42	0	***1***	x	x	-	x	x	x
RD_MSG_LEN	0x44	0	***1***	x	x	-	x	x	x
READ_ID	0x48	0	***1***	x	x	-	x	x	x
DIG_READ	0xA2	0	***1***	x	x	-	x	x	x
DIG_WRITE	0xA0	0	***1***	x	x	-	x	x	x
CHECK_SUM	0xF2	0	***1***	x	x		0	0	0
ERASE_MEM	0x24	0	***1***	x	x	-	0	0	0
CHIP_ERASE	0x26	0	***1***	x	x	-	0	0	0
PWR_UP	0x10	***1***	0	x	x	-	x	x	x
PWR_DN	0x12	0	***1***	x	x	-	x	x	x
WR_CFG_REG	0x80	0	***1***	x	x	-	x	x	x
RD_CFG_REG	0x90	0	***1***	x	x	-	x	x	x

10.1 Audio Play and Record Commands

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

10.1.1 Play Voice Prompt

PLAY_VP				
Byte Sequence:	Host controller	0xA6	*Index*[15:8]	Index[7:0]
	ISD61S00	Status Byte	Status Byte	Status Byte
Description:	Play Voice Prompt *Index*			
Interrupt Generation:	CMD_ERR if not accepted. CMD_FIN when playback is finished.			

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

.

10.1.2 Play Voice Prompt @Rn, n = 0 ~ 7

PLAY_VP@Rn				
Byte Sequence:	Host controller	0xAE	*n = 0 ~ 7*	
	ISD61S00	Status Byte	Status Byte	
Description:	Play Voice Prompt, *Index@Rn*			
Interrupt Generation:	CMD_ERR if not accepted. CMD_FIN when playback is finished.			

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

- R0[7:0] = CFG20, R0[15:8] = CFG21
- R1[7:0] = CFG22, R1[15:8] = CFG23
- R2[7:0] = CFG24, R2[15:8] = CFG25
- R3[7:0] = CFG26, R3[15:8] = CFG27
- R4[7:0] = CFG28, R4[15:8] = CFG29
- R5[7:0] = CFG2A, R5[15:8] = CFG2B
- R6[7:0] = CFG2C, R6[15:8] = CFG2D
- R7[7:0] = CFG2E, R7[15:8] = CFG2F

10.1.3 Play Voice Prompt, Loop

PLAY_VP_LP						
Byte Sequence:	Host controller	0xA4	*Index*[15:8]	Index[7:0]	LoopCnt[15:8]	LoopCnt[7:0]

	ISD61S00	Status Byte	Status Byte	Status Byte	Status Byte	Status Byte
Description:	Play Voice Prompt *Index, Loop*					
Interrupt Generation:	CMD_ERR if not accepted. CMD_FIN when playback is finished.					

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

新唐科技 NUVOTON INTELLECTUAL PROPERTY

10.1.4 Play Voice Prompt, Loop, @Rn, n = 0 ~ 7

PLAY_VP_LP@Rn					
Byte Sequence:	Host controller	0xB2	*n = 0 ~ 7*	LoopCnt[15:8]	LoopCnt[7:0]
	ISD61S00	Status Byte	Status Byte	Status Byte	Status Byte
Description:	Play Voice Prompt, Loop, *Index@Rn*				
Interrupt Generation:	CMD_ERR if not accepted. CMD_FIN when playback is finished.				

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

- R0[7:0] = CFG20, R0[15:8] = CFG21
- R1[7:0] = CFG22, R1[15:8] = CFG23
- R2[7:0] = CFG24, R2[15:8] = CFG25
- R3[7:0] = CFG26, R3[15:8] = CFG27
- R4[7:0] = CFG28, R4[15:8] = CFG29
- R5[7:0] = CFG2A, R5[15:8] = CFG2B
- R6[7:0] = CFG2C, R6[15:8] = CFG2D
- R7[7:0] = CFG2E, R7[15:8] = CFG2F

10.1.5 Stop Loop-Play Command

STOP_LP				
Byte Sequence:	Host controller	0x2E		
	ISD61S00	Status Byte		
Description:	Stop current loop-play command			
Interrupt Generation:	Command itself does not generate interrupt, only those commands that it is stopping.			

This command stops any current PLAY_VP_LP or PLAY_VP_LP@Rn command active in the ISD61S00. The STOP_LP command does not flush the audio command buffer; that is, any command queued in the buffer when a STOP_LP is issued will be executed thereafter. When device has finished the active command a CMD_FIN interrupt will be generated. If there is no active PLAY_VP_LP or PLAY_VP_LP@Rn command then STOP_LP will have no effect.

10.1.6 Execute Voice Macro

EXE_VM				
Byte Sequence:	Host controller	0xB0	*Index*[15:8]	Index[7:0]
	ISD61S00	Status Byte	Status Byte	Status Byte
Description:	Play voice macro *Index*			

Interrupt Generation:	CMD_ERR if not accepted. CMD_FIN when playback is finished.

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

10.1.7 Execute Voice Macro @Rn, n = 0 ~ 7

EXE_VM@Rn				
Byte Sequence:	Host controller	0xBC	*n = 0 ~ 7*	
	ISD61S00	Status Byte	Status Byte	
Description:	Play voice macro *Index@Rn*			
Interrupt Generation:	CMD_ERR if not accepted. CMD_FIN when playback is finished.			

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

- R0[7:0] = CFG20, R0[15:8] = CFG21
- R1[7:0] = CFG22, R1[15:8] = CFG23
- R2[7:0] = CFG24, R2[15:8] = CFG25
- R3[7:0] = CFG26, R3[15:8] = CFG27
- R4[7:0] = CFG28, R4[15:8] = CFG29
- R5[7:0] = CFG2A, R5[15:8] = CFG2B
- R6[7:0] = CFG2C, R6[15:8] = CFG2D
- R7[7:0] = CFG2E, R7[15:8] = CFG2F

10.1.8 Record Message

REC_MSG				
Byte Sequence:	Host controller	0x38		
	ISD61S00	Status Byte		
Description:	Initiates a managed record at first available location in memory.			
Interrupt Generation:	CMD_ERR if not accepted. CMD_FIN when recording complete. FULL_ERR when device fills available memory.			

This command initiates a record operation. Before execution of command a valid signal path must be set up and the device must have space in the audio command buffer. If device is or becomes full, an interrupt is generated and the FULL_ERR bit of the interrupt status register is set. Recording is terminated by

issuing a STOP command. After the operation is complete the begin address of the message can be read, along with the number of sectors recorded, with the READ_MSG_ADDR command.

The command will be accepted if status bits PD=0, DBUF_RDY=1, VM_BSY=0 and CBUF_FUL=0. If any of these conditions are not met then a CMD_ERR interrupt will be generated and the command ignored. If memory becomes full while recording a FULL_ERR interrupt will be generated. If device was full before the record was sent then a FULL_ERR interrupt will be generated and READ_MSG_ADDR will return a length of zero. When a record is terminated by a stop command a CMD_FIN interrupt will be generated once the recording process is complete.

Note, if REC_MSG_SEC_LENGTH[15:0] is 0 (CFG_REG5E= REC_MSG_SEC_LENGTH[15:8]; CFG_REG5D= REC_MSG_SEC_LENGTH[7:0]), the REC_MSG will need CMD STOP to end. If REC_MSG_SEC_LENGTH[15:0] is NOT 0, the REC_MSG command will automatically end after the number of sectors which set by REC_MSG_SEC_LENGTH[15:0] have been programmed

10.1.9 Record Message at Address

REC_MSG@					
Byte Sequence:	Host controller	0x3A	A[23:16]	A[15:8]	A[7:0]
	ISD61S00	Status Byte	Status Byte	Status Byte	Status Byte
Description:	Initiate a managed record starting at sector address A/4096				
Interrupt Generation:	CMD_ERR if not accepted. CMD_FIN when recording complete. ADDR_ERR if invalid address sent.				

This command initiates a record operation starting at a specified address. Before execution of command a valid signal path must be set up and the device must have space in the audio command buffer. If device is or becomes full, the RM_FULL bit of the status register is set. Recording is terminated by issuing a STOP command. After the operation is complete the begin address of the message can be read, along with the number of sectors recorded out with the READ_MSG_ADDR command.

The command will be accepted if status bits PD=0, DBUF_RDY=1, VM_BSY=0 and CBUF_FUL=0. If any of these conditions are not met then a CMD_ERR interrupt will be generated and the command ignored. If command is terminated after the command byte is sent no interrupt will be generated. If memory becomes full while recording a FULL_ERR interrupt will be generated. If device was full before the record was sent then a FULL_ERR interrupt will be generated and READ_MSG_ADDR will return a length of zero. If the address sent is not a blank sector then an ADDR_ERR interrupt is generated. When a record is terminated by a stop command a CMD_FIN interrupt will be generated once the recording process is complete.

Note, if REC_MSG_SEC_LENGTH[15:0] is 0 (CFG_REG5E= REC_MSG_SEC_LENGTH[15:8]; CFG_REG5D= REC_MSG_SEC_LENGTH[7:0]), the REC_MSG will need CMD STOP to end. If REC_MSG_SEC_LENGTH[15:0] is NOT 0, the REC_MSG command will automatically end after the number of sectors which set by REC_MSG_SEC_LENGTH[15:0] have been programmed.

10.1.10Play Message at Address

PLAY_MSG@							
Byte	Host controller	0x3C	A[23:16]	A[15:8]	A[7:0]	OFF[15:8]	OFF[7:0]

Sequence:	ISD61S00	Status Byte	Status Byte	Status Byte	Status Byte	Status Byte	Status Byte
Description:	Initiate a managed record starting at sector address A/4096 + OFF						
Interrupt Generation:	CMD_ERR if not accepted. CMD_FIN when playback complete. ADDR_ERR if invalid address sent.						

This command initiates a play of a recorded message starting at a specified address, with a specified sector offset. Before execution of command a valid signal path must be set up and the device must have space in the audio command buffer. If an address is sent that is not a valid message an error interrupt is generated with the ADDR_ERR bit set. Playback can be terminated by issuing a STOP command. After completion of playback, the device will generate an interrupt. This command can be used to randomly access a message at any 4K sector boundary by sending the appropriate offset. The bottom 12 bits of the start address are ignored as messages must begin at a 4Kbyte sector boundary. If command is sent with less than five bytes of data the command is ignored.

The command will be accepted if status bits PD=0, DBUF_RDY=1, VM_BSY=0 and CBUF_FUL=0. If any of these conditions are not met then a CMD_ERR interrupt will be generated and the command ignored. If command is terminated after the command byte is sent no interrupt will be generated. Once playback is finished a CMD_FIN interrupt will be generated. If the address sent is not a beginning of message sector then an ADDR_ERR interrupt will be generated. If the offset is greater than the message length then an ADDR_ERR interrupt will be generated.

10.1.11Play Silence

PLAY_SIL				
Byte Sequence:	Host controller	0xA8	*LEN*[7:0]	
	ISD61S00	Status Byte 0	Status Byte 0	
Description:	Play silence for LEN*32ms			
Interrupt Generation:	CMD_ERR if not accepted. CMD_FIN when silence playback complete.			

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

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

10.1.12Stop Command

STOP				
Byte Sequence:	Host controller	0x2A		
	ISD61S00	Status Byte		

Description:	Stop current audio command and flush command buffer.
Interrupt Generation:	Command itself does not generate interrupt, only those commands that it is stopping.

This command stops any current audio command active in the ISD61S00. If a PLAY_MSG@, PLAY_VP, EXE_VM or PLAY_SIL command is active playback is stopped immediately. If a REC_MSG command is active recording is stopped at the next available byte boundary. The STOP command flushes the audio command buffer, that is any command queued in the buffer when a STOP is issued will not be executed. When device has finished the active command a CMD_FIN interrupt will be generated. STOP will not stop an ERASE_MSG or ERASE_MEM operation. If there is no active command then STOP will have no effect.

10.1.13Erase Message at Address

ERASE_MSG@					
Byte Sequence:	Host controller	0x3E	A[23:16]	A[15:8]	A[7:0]
	ISD61S00	Status Byte	Status Byte	Status Byte	Status Byte
Description:	Erase message starting at sector address A/4096				
Interrupt Generation:	CMD_ERR if not accepted. CMD_FIN when erase complete. ADDR_ERR if invalid address sent.				

This command erases the message starting at the specified address. The command will be accepted if status bits PD=0, DBUF_RDY=1, VM_BSY=0, CBUF_FUL=0 and CMD_BSY=0. If any of these conditions are not met then a CMD_ERR interrupt will be generated and the command ignored. If address sent is not a beginning of message sector an ADDR_ERR interrupt is generated. Upon completion of command CMD_FIN interrupt is generated. While the device is erasing no other commands will execute. If a PLAY or REC is sent it is queued in the command buffer and will not execute until the erase is finished. If a DIG_RD or DIG_WR command is sent to the device, RDY/BSYB will hold off any data transfer until the ERASE_MSG@ has completed.

10.1.14SPI Send Audio Data

SPI_SND_AUD							
Byte Sequence:	Host controller	0Xaa	D0[7:0]	D0[15:8]	….	Dn[7:0]	Dn[15:8]
	ISD61S00	Status Byte					
Description:	Write audio data via SPI interface.						
Interrupt Generation:	OVF_ERR if RDY/BSY violated.						

This command allows the user to send audio data, in 16bit PCM format, down the SPI interface for feed-through or recording. Before execution of command a valid signal path must be set up.

If recording data (SPI record), then: a signal path must be set up for SPI input to the compressor. A valid record command is then sent followed by the SPI_SND_AUD command. Multiple SPI_SND_AUD commands can be issued to write data to the ISD61S00. To finish recording a STOP command is sent and device will respond with a CMD_FIN interrupt. If a memory overflow occurs during the operation a

FULL_ERR interrupt will be generated and no more data will be accepted. RDY/BSYB will handshake dataflow if device cannot compress and write data at the rate sent down the SPI interface.

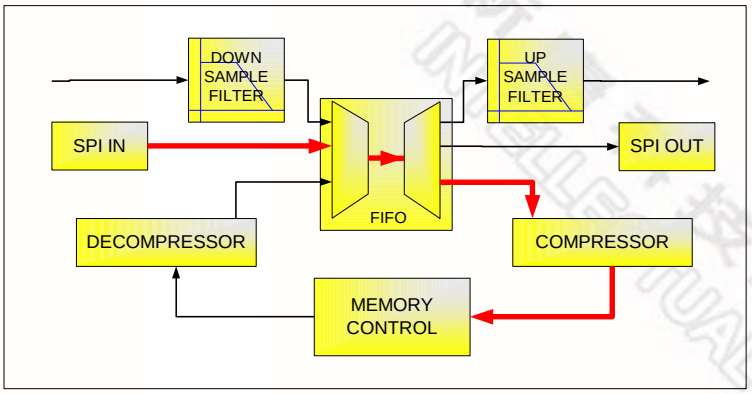


Figure 10-1 SPI Record

If sending audio data to the analog outputs (SPI feed-through), then: (1) the path must be set up for SPI input to the signal path. (2) A SPI_SND_AUD command is then sent. Multiple SPI_SND_AUD commands can be issued to write data to the ISD61S00. (3) To finish sending audio data a STOP command is sent and device will respond with a CMD_FIN interrupt. RDY/BSYB will handshake dataflow to the sample rate set by the audio configuration register. If host cannot keep up with data rate audio will be corrupt. Once audio data is sent, raise SSB high and device will continue to play zero samples out the signal path until reconfigured or more data is sent.

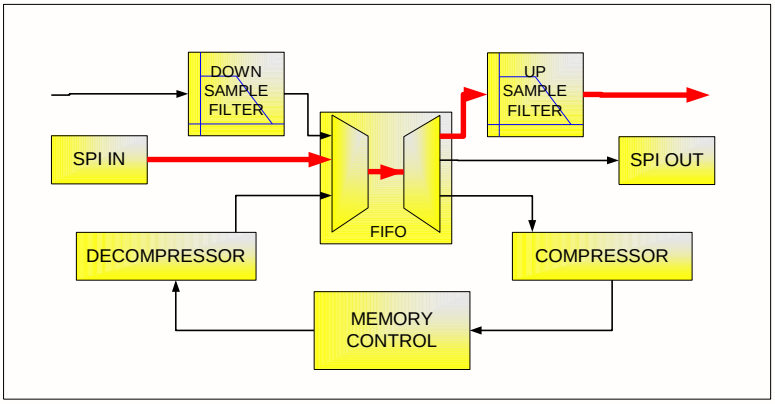


Figure 10-2 SPI Feed-through

The RDY/BSYB signal will go low whenever the internal FIFO is full. If no path or record operation is set up then RDY/BSYB will not return high until command is terminated. If RDY/BSYB is ignored then an OVF_ERR interrupt is generated.

10.1.15 SPI Receive Audio Data

SPI_RCV_AUD

Byte Sequence:	Host controller	0xAC					
	ISD61S00	Status Byte	D0[7:0]	D0[15:8]	….	Dn[7:0]	Dn[15:8]
Description:	Read audio data via SPI interface.						
Interrupt Generation:	OVF_ERR if RDY/BSY violated.						

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

If receiving recorded audio data (SPI playback), then: a signal path must be set up for SPI output from the compressor. A valid play command is then sent followed by the SPI_RCV_AUD command. Multiple SPI_RCV_AUD commands can be sent. To finish receiving data a STOP command is sent and device will generate a CMD_FIN interrupt. When the end of message is reached a CMD_FIN interrupt will be generated and zero will be sent as data.

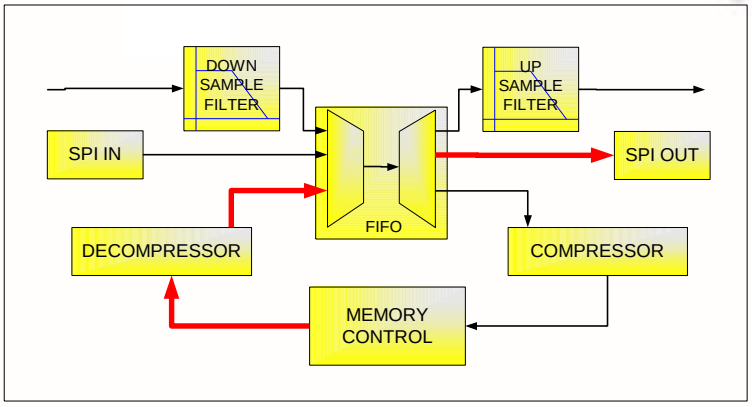


Figure 10-3 SPI Playback

If reading audio data from the analog inputs (feed-through SPI) then: (1) the path must be set up for SPI output from the signal path. (2) A SPI_RCV_AUD command is then sent. Multiple SPI_RCV_AUD commands can be sent. (3) To finish receiving data a STOP command is sent and device will generate a CMD_FIN interrupt. RDY/BSYB will handshake dataflow to the sample rate set by the audio configuration register. If host cannot keep up with data rate, audio will be corrupt.

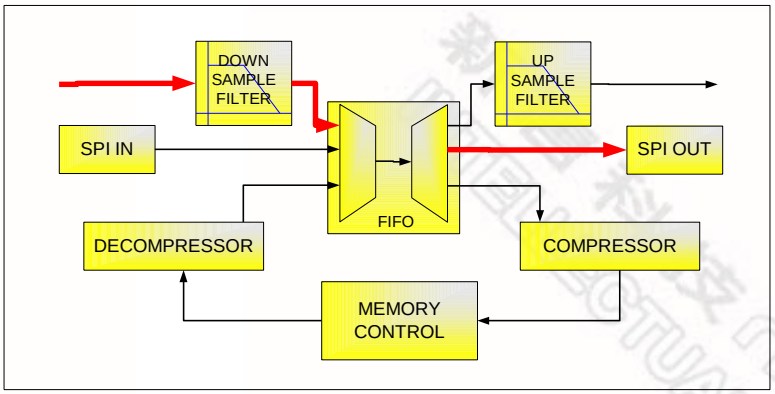


Figure 10-4 Feed-through SPI

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

10.1.16SPI Send Compressed Audio Data for direct programming to flash

SPI_SND_CMPR							
Byte Sequence:	Host controller	0XDC	D0[7:0]	D0[15:8]	….	Dn[7:0]	Dn[15:8]
	ISD61S00	Status Byte					
Description:	During REC_MSG or REC_MSG@, Write compressed data via SPI interface to direct program the compressed data to flash memory						
Interrupt Generation:	OVF_ERR if RDY/BSY violated.						

This command allows the user to send compressed audio data, down the SPI interface for recording. Before execution of command make sure CFG_REG2=0x00 (ENC, DEC, SPI_IN, SPI_OUT should be all 0)

A valid REC_MSG or REC_MSG@ command is then sent followed by the SPI_SND_CMPR command. Multiple SPI_SND_CMPR commands can be issued to write data to the ISD61S00. To finish recording a STOP command is sent and device will respond with a CMD_FIN interrupt. If a memory overflow occurs during the operation a FULL_ERR interrupt will be generated and no more data will be accepted. RDY/BSYB will handshake dataflow.

Sample Sequence:

1. Make sure CFG_REG2=0x00
2. Issue REC_MSG or REC_MSG@ command

3. Issue SPI_SND_CMPR with compressed data
4. STOP

10.1.17 SPI Receive previous REC_MSG or REC_MSG@ compressed Audio Data stored in flash

SPI_RCV_CMPR							
Byte Sequence:	Host controller	0xBE					
	ISD61S00	Status Byte	D0[7:0]	D0[15:8]	….	Dn[7:0]	Dn[15:8]
Description:	During PLAY_MSG or PLAY_MSG@, read previous REC_MSG or REC_MSG@ compressed Audio data stored in flash via SPI						
Interrupt Generation:	OVF_ERR if RDY/BSY violated.						

This command allows the user receiving previous REC_MSG or REC_MSG@ compressed Audio Data stored in flash through SPI interface. Before execution of command make sure CFG_REG2=0x00 (ENC, DEC, SPI_IN, SPI_OUT should be all 0)

A valid PLAY_MSG@ command is then sent followed by the SPI_RCV_CMPR command. Multiple SPI_RCV_CMPR commands can be sent. To finish receiving data a STOP command is sent and device will generate a CMD_FIN interrupt. When the end of message is reached a CMD_FIN interrupt will be generated.

Sample Sequence:

1. Make sure CFG_REG2=0x00
2. Issue PLAY_MSG@ command with correct start address
3. Issue SPI_RCV_CMPR and read compressed data from MISO
4. STOP or wait CMD_FIN interrupt

10.1.18 SPI Send Compressed Data to Decode

SPI_SND_DEC						
Byte Sequence:	Host controller	0xC0	D0[7:0]	D1[7:0]	….	Dn[7:0]
	ISD61S00	Status Byte				
Description:	Write compressed audio data via SPI interface.					
Interrupt Generation:	OVF_ERR if RDY/BSYB violated.					

This command allows the user to send compressed audio data, in a byte formatted bit stream, down the SPI interface to the de-compressor and signal path. Before execution of command a valid signal path must be set up. Valid paths are similar to a standard playback. Multiple SPI_SND_DEC commands can be issued to send data to the ISD61S00. To finish decoding a STOP command is sent and device will respond with a CMD_FIN interrupt. RDY/BSYB will handshake dataflow if device cannot accept any further data for decompression. If host cannot keep up with data rate audio output will be corrupted.

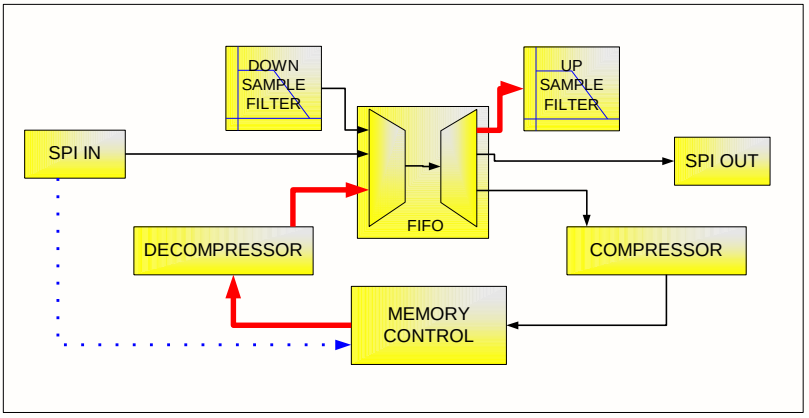


Figure 10-5 SPI Send Compressed Data to Decode

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

10.1.19 SPI Receive Encoded Data

SPI_RCV_ENC						
Byte Sequence:	Host controller	0xC2				
	ISD61S00	Status Byte	D0[7:0]	D1[7:0]	….	Dn[7:0]
Description:	Read compressed audio data via SPI interface.					
Interrupt Generation:	OVF_ERR if RDY/BSYB violated.					

This command allows the user to receive compressed audio data, in a byte formatted bit stream, from the SPI interface for use or storage outside the ISD61S00. Before execution of command a valid recording signal path must be set up such that compressor is active. Multiple SPI_RCV_ENC commands can be sent to receive compressed data from the ISD61S00. To finish receiving data a STOP command is sent and device will generate a CMD_FIN interrupt.

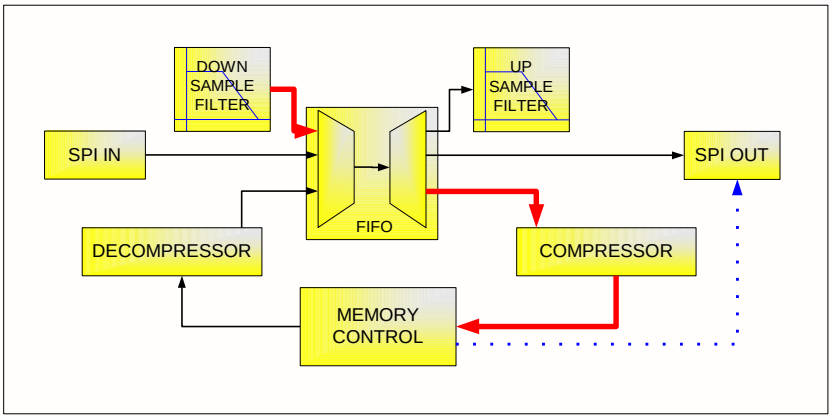


Figure 10-6 SPI Received Encoded Data

RDY/BSYB will handshake dataflow to the sample rate and compression bit rate set by the audio configuration register. If host cannot keep up with data rate compressed audio will be corrupt.

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

10.2 Device Status Commands.

10.2.1 Read Status

Powered up:

READ_STATUS					
Byte Sequence:	Host	0x40	0xXX	0xXX	0xXX
	ISD61S00	Status Byte	Interrupt Status Byte 1	Interrupt Status Byte 2	Interrupt Status Byte 3
Description:	Query device status.				

Powered down:

READ_STATUS			
Byte Sequence:	Host controller	0x40	0xXX
	ISD61S00	Status Byte 80h	00h
Description:	Query device status.		

This command queries the ISD61S00 device status. For details of device status see Section 7.1. If device is powered up, the four status bytes will be repeated for each four dummy bytes sent to the SPI interface. If device is powered down, only one status byte 80h shows up to the SPI interface at the same time the command is sent. This command is always accepted.

10.2.2 Read Interrupt

READ_INT					
Byte Sequence:	Host	0x46	0xXX	0xXX	0xXX
	ISD61S00	Status Byte	Interrupt Status Byte 1	Interrupt Status Byte 2	Interrupt Status Byte 3
Description:	Query device status and clear interrupt flags.				

This command queries the ISD61S00 device status and clears any pending interrupts. After this command the hardware interrupt line will return inactive. The INT bit of the status register along with any status error bits will return inactive. This command is accepted whenever device is powered up.

10.2.3 Read Recorded Message Address Details

RD_MSG_ADD							
Byte Sequence:	Host	0x42	X	X	X	X	X
	ISD61S00	STATUS0	A[23:16]	A[15:8]	A[7:0]	LEN[15:8]	LEN[7:0]
Description:	Reports the start sector address A/4096 of current message and length, LEN, in 4kByte sectors						

This interrogates the status of the last or current audio record command. It returns the start address so that the user can address a message for playback and also returns the number of sectors that the message has used. It should be issued immediately after a record is initiated to correctly get retrieve the message start address and current length of message. After a STOP command or overflow condition, data is valid for final message length after CMD_BSY has returned low and before a subsequent audio command is issued.

10.2.4 Read Message Length

RD_MSG_LEN				
Byte Sequence:	Host controller	0x44	0xXX	0xXX
	ISD61S00	Status Byte	LEN[15:8]	LEN[7:0]
Description:	Read number of sectors played by current PLAY command.			

This command returns the number of sectors played by the current PLAY_MSG@ command, or recorded by the current REC_MSG command. This command is used to determine the offset position of the currently playing/recording message. It can be used to resume the playback of a message at a particular sector. For instance, if a PLAY_MSG(SA, 0) command was sent to start playback of a message from SA then a STOP was sent during playback of the third sector of the message, then RD_MSG_LEN would return LEN=3. A subsequent PLAY_MSG(SA,2) command would restart the playback from the beginning of the sector where playback was stopped; that is send PLAY_MSG(SA, LEN-1). Now, if a STOP was issued after an additional three sectors of playback (message is now playing the sixth sector), RD_MSG_LEN would return LEN=6.

10.2.5 Read ISD61S00 ID

READ_ID						
Byte Sequence:	Host controller	0x48	0xXX	0xXX	0xXX	0xXX
	ISD61S00	Status Byte	PART_ID	MAN_ID	MEM_TYPE	DEV_ID
Description:	Return memory ID of internal memory					

This command queries the ISD61S00 to returns four bytes to indicate the ISD61S00 family member and the manufacturer, size and type of internal memory of the device. The four bytes returned are:

PART_ID – Identifies which ISD61S00 family member.

MAN_ID – Manufacturer ID, which is 0xEF for Winbond.

MEM_TYPE – Memory type, which is 0x30.

DEV_ID – Device ID indicates the memory size as the table below.

Capacity	Value
4Mb	13
8Mb	14
16Mb	15

32Mb	16
64Mb	17

10.3 Digital Memory Commands.

This section describes the 4 digital data commands that can be sent to the device. Digital commands are ones that read, write or erase data directly in the flash memory through a separate interface than the audio data command interface. Digital memory commands other than erase, can occur simultaneously with audio memory commands.

10.3.1 Digital Read

DIG_READ								
Byte Sequence:	Host controller	0xA2	A[23:16]	A[15:8]	A[7:0]	0xXX	…	0xXX
	ISD61S00	Status	Status	Status	Status	D0	…	Dn
Description:	Initiates a digital read of memory from address A[23:0].							
Interrupt Generation:	ADDR_ERR if memory protected or RDY/BSYB violated. OVF_ERR if read past end of array.							

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

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

10.3.2 Digital Write

DIG_WRITE								
Byte Sequence:	Host controller	0xA0	A[23:16]	A[15:8]	A[7:0]	D0	…	Dn
	ISD61S00	Status	Status	Status	Status	Status	…	Status
Description:	Initiates a digital write to memory from address A[23:0].							
Interrupt Generation:	ADDR_ERR if memory protected or RDY/BSYB violated. OVF_ERR if write past end of array.							

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

The command will always be accepted and RDY/BSYB will go low until any active digital memory command is complete. If a digital write is attempted in write protected memory, data will be ignored and an ADDR_ERR interrupt will be generated. If a write is attempted past the end of memory an OVF_ERR interrupt will be generated. If RDY/BSYB is violated then data will ignored and an ADDR_ERR interrupt will be generated. Once the SPI transaction has ended the ISD61S00 will finish the flash write operation. When this operation is complete the ISD61S00 will generate a WR_FIN interrupt. While device is actively writing to flash memory the CMD_BSY bit will be active.

10.3.3 Erase Memory

ERASE_MEM								
Byte Sequence:	Host controller	0x24	SA[23:16]	SA[15:8]	SA[7:0]	EA[23:16]	EA[15:8]	EA[7:0]
	ISD61S00	Status	Status	Status	Status	Status	Status	Status
Description:	Erases memory from sector containing SA to sector containing EA.							
Interrupt Generation:	ADDR_ERR if memory protected. CMD_ERR if device is busy. CMD_FIN when erase operation complete.							

This erases memory from the sector containing start address SA to the sector containing end address EA. The minimum erase block of internal memory is a 4kByte sector.

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

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

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

10.3.4 Chip Erase

CHIP_ERASE			
Byte Sequence:	Host controller	0x26	0x01
	ISD61S00	Status Byte	Status Byte
Description:	Initiate a mass erase of memory.		

Interrupt Generation:	CMD_ERR if device is busy and cannot accept command. CMD_FIN when erase operation complete.

This erases the entire contents of the internal memory.

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

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

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

10.3.5 CHECKSUM

CHECKSUM					
Byte Sequence:	Host controller	0xF2	EA[23:16]	EA[15:8]	EA[7:0]
	ISD61S00	Status Byte	Status	Status	Status
Description:	Initiate a checksum of memory.				
Interrupt Generation:	CMD_ERR if device is busy and cannot accept command. CMD_FIN when erase operation complete.				

This initiates a 4-byte checksum calculation from the very beginning to the specified end address. The calculated checksum is stored in register 0x59[7:0] 0x5A[15:8] 0x5B[23:16] 0x5C[31:24]. To re-calculate the checksum with a different end address, users have to set register 0x18 bit-0 to clear the registers 0x59 ~ 0x5C.

The command will be accepted if status bits PD=0, DBUF_RDY=1, VM_BSY=0, CBUF_FUL=0 and CMD_BSY=0

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

10.4 Device Configuration Commands.

This section describes commands used to configure the ISD61S00. These commands are used to:

- Set up the clocking regime of the device including clock source and setting the master sample rate.
- Configure the audio signal path.
- Control other processing blocks within the device.
- Configure the compression and sample rate for message recording.

10.4.1 PWR_UP – Power up

PWR_UP				
Byte Sequence:	Host controller	0x10		
	ISD61S00	Status		…
Description:	Powers up device and initiates the power up sequence.			

This command powers up the device. If device already powered up this command has no effect. If powered down, then the internal power up sequence is initiated. If the PU voice macro is present this is executed, otherwise the device defaults to previous clock configuration values or the default values following a reset condition. When power up is complete the PD bit of the status register will go low and the RDY bit high. Until this event no other commands will be accepted by the ISD61S00.

A formal power-up procedure is as follows:

- Send PWR_UP command.
- Poll Status until bit-6 DBUF_RDY goes high, indicates device is ready to accept commands.
- Poll Status until bit-2 VM_BSY goes low, indicates voice macro 1 is finished.

10.4.2 PWR_DN – Power Down

PWR_DN				
Byte Sequence:	Host controller	0x12		
	ISD61S00	Status		…
Description:	Powers down the device after any active commands finish			

This command powers down the device. If the device is currently executing a command the device will powers down when the command finishes. If sent while recording, without a STOP command sent first , then device will record until full then power down. If playing or executing a voice macro, device will power down after playback is finished. The PWR_DN command will not generate an interrupt. PWR_DN has executed when PD bit of status goes high.

10.4.3 WR_CFG_REG – Write Configuration Register

WR_CFG_REG						
Byte Sequence:	Host controller	0x80 | REG[9:8]	REG[7:0]	D0	…	Dn
	ISD61S00	STATUS0			…	
Description:	Loads configuration register CFG[REG] with D0. Data bytes 1..n can be sent to load CFG[REG+1] with D1 to CFG[REG+n] with Dn.					

This command loads configuration registers starting at the address specified. If multiple data bytes are sent, additional configuration registers are loaded. See Section 8 for details on configuration registers.

10.4.4 RD_CFG_REG – Read Configuration Register

RD_CFG_REG

Byte Sequence:	Host controller	0x90 | REG[9:8]	REG[7:0]	X	…	X
	ISD61S00	STATUS0		D0	…	Dn
Description:	Reads configuration register CFG[REG] and outputs to SPI as D0. Data bytes 1..n can be read sequentially from CFG[REG+1] to CFG[REG+n].					

This command reads the configuration register starting at the address specified. If multiple data bytes are sent, additional configuration registers are read. See Section 8 for details on configuration registers.

10.5 Device Power Up Sequence

It is important to sequence the power up configuration of the device to prevent unwanted transients from occurring on audio outputs of the device. A general guideline to achieve this is:

4. Send PWR_UP command and wait until device ready.
5. Send correct clock configuration if not using default values or if not set in PU voice macro.
6. Power up CODEC.
7. Set mixer path with DAC/ADC muted.
8. Power up analog path: power up drivers and DAC with outputs muted, then unmute outputs.
9. Unmute DAC/ADC.

A general power down sequence is the reverse of above.

11. ISD61S00 MEMORY MANAGEMENT

The ISD61S00 employs several memory management techniques to make audio recording and playback transparent to the host controller. The address space of the ISD61S00 starts at address zero of the internal memory. This internal memory size can range from 2M to 64M with an erasable sector size of 4kBytes. The following sections will describe the ISD61S00 memory management architecture and the message management functions.

11.1 ISD61S00 Memory Format

The Recording Memory Pointer (RMP) divides the ISD61S00 memory address space into two blocks, Reserved Memory and Recording Memory. The RMP is a two-byte address pointer pointing to a 4kByte memory sector which is the first sector available to users for recording messages. Memory between address zero and the RMP pointer is considered the Reserved Memory for pre-recorded audio (Voice Prompts), pre-programmed macro scripts (Voice Macros), digital read/write access for other applications (User Data) and memory sectors reserved for the first sector of the message recordings and re-recordable pre-recorded messages (Reserved Sectors). The memory between the RMP and the end of memory is considered the Recording Memory allocated for recording messages (Message Recordings).

Figure 11-1 illustrates the memory format with the RMP set at address 4000h or sector 4. Section 11.3 will provide more detail regarding the setting of the RMP.

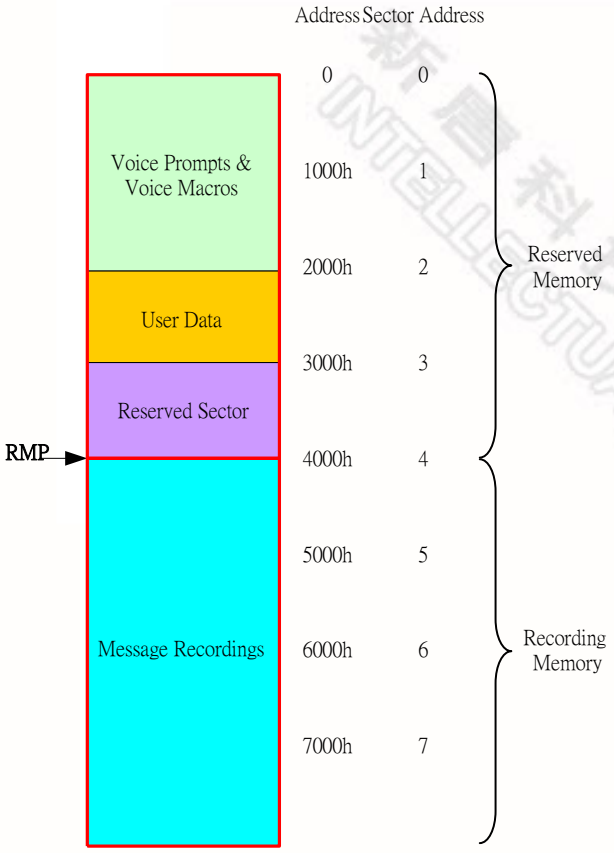


Figure 11-1 ISD61S00 Memory Format

11.2 Message Management

The message management schemes implemented on the ISD61S00 are:

1. Voice Prompts: A collection of pre-recorded audio that can be played back using the PLAY_VP SPI command or Voice Macros.
2. Voice Macros: A powerful voice script allowing users to create custom macros to play Voice Prompts, play message recordings, insert silence and configure the device. Voice Macros are executed with a single SPI command.
3. User Data: Memory sectors defined and allocated by the users for use in other applications
4. Reserved Sectors: Memory sectors reserved for the first sector of the message recordings (Empty Message) and re-recordable pre-recorded messages (Re-recordable Message).
5. Message Recordings: Messages recorded, played and erased “on the fly” by the users.

11.2.1 Voice Prompts

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

1. An index pointing to the pre-recorded audio
2. Pre-recorded audio

To play a Voice Prompt, the ISD61S00 use the index of the Voice Prompt to locate and play the pre-recorded audio. This approach allows users to easily manage the pre-recorded audio without the need to update the code on the host controller. In addition, the users can store a multitude of pre-recorded audio without the overhead of maintaining a complicated lookup table. To assist customers in creating the Voice Prompts, a software tool, the ISD61S00 Voice Prompt Editor and writer are available for development purposes.

11.2.2 Voice Macros

Voice Macros are a powerful voice script that allows users to customize their own play patterns such as play Voice Prompts, play message, insert silence, change the master sample clock, power-down the device and configure the signal path, including gain and volume control. Voice Macros are executed using a single SPI command and are accessed using the same index structure as Voice Prompts. This means that a Voice Macro (or Voice Prompt) can be updated on the ISD61S00 without the need to update code on the host micro-controller since absolute addresses are not needed.

The following locations have been reserved for three special Voice Macros:

Index 0: Power-On Initialization (POI)

Index 1: Power-Up (PU)

Index 2: GPIO WakeUp

These Voice Macros allow the users to customize the ISD61S00 power-on and power-up procedures and are executed automatically when utilized. The built-in voice macros will be used when they are not utilized. Please see Section 12 for details.

Here is an example to illustrate the usage of the PU Voice Macro.

- WR_CFG_REG(0x05,0x01)
- WR_CFG_REG (0x06,0x02)
- FINISH

The above PU Voice Macro will perform the following:

- Set up a feed-through path from ANAIN to AUDOUT

The following is the complete list of the commands for Voice Macro:

- WR_CFG_REG(reg,*n*) – Set configuration register reg to value *n*.
- PWR_DN – Power down the ISD61S00.
- PLAY_VP(*i*) – Play Voice Prompt index *i*.
- PLAY_VP@Rn – Play Voice Prompt at the index *i* contained in Register n.
- PLAY_VP_LP(*I*) – Play Voice Prompt index *I* with looping
- PLAY_VP@Rn_LP – Play Voice Prompt at the index *i* contained in Register n with looping
- PLAY_MSG@(*sec-addr*) – Play message starting at sector address *sec-addr*

- PLAY_SIL(*n*) – Play silence for *n* units. A unit is 32ms at master sampling rate of 32 kHz.
- WAIT_INT – Wait until current play command finishes before executing next macro instruction.
- EXE_VM – Put EXE_VM in a Voice Macro will force to jump to this VM
- EXE_VM@Rn – Put EXE_VM@Rn in a Macro will force to jump to this VM@Rn
- FINISH – Finish the voice macro and exit.

11.2.3 User Data

User Data consist of 4kByte chunks of erasable sectors defined and allocated by the users for use in other applications. The users have the freedom not to allocate or reserve any memory sectors. A software tool, the ISD61S00 Voice Prompt Editor is available to assist customers in allocating such memory.

11.2.4 Reserved Sectors

Reserved sectors consist of 4kByte chunks of erasable sectors reserved for the Empty Messages and Re-recordable Messages. The Empty Message is an empty sector made of a 4kByte memory sector reserved for message recordings. A SPI record command pointing to the Empty Message starts the message recording into the Empty Message and will continue to record messages in the free memory located in Recording Memory if the message recording is over 4kBytes. The users can send a play, erase or record command pointing to the Empty Message to play, erase the entire message or re-record a brand new message.

The Re-recordable Message, a 4kByte memory sector contains message recordings and a sector address pointer (see Section 11.3 for details) pointing to the message sector located in Recording Memory. Unlike the Empty Message where no message has been recorded, the users can play back a complete message by issuing a SPI play command pointing to the Re-recordable Message. The users can also send an erase or record command pointing to the Re-recordable Message to erase the entire message or re-record a brand new message.

11.2.5 Message Recordings

Message Recordings are messages that can be recorded, played and erased “on the fly” by the users. These messages are recorded in 4kByte chunks of erasable sectors with each message sector containing a header pointing to the next message sector. The ISD61S00’s unique message management architecture allows users to record messages without specifying an address. The ISD61S00 will start recording at the first available sector in Recording Memory and continue to record into free memory until memory is full or a STOP command is issued. Upon completion of the record operation, the users need to read back the message sector start address for subsequent playback.

To playback the message recordings, the users need to send a play command pointing to the message sector start address. For partial message playback, the users can send a play command with a sector offset.

To erase the message recordings, the users need to send an erase command pointing to the message sector start address.

A list of commands for Message Recordings are detailed below:

- REC_MSG – starts a record operation.
- STOP – stops current record operation.
- READ_STATUS – wait until record operation is finished as indicated by the CMD_FIN bit.
- READ_INT – clear the interrupt.

- READ_MSG_ADD – This command returns a three byte starting address (*A*) of the message along with a two byte sector length. Use *A* to address the message in subsequent playback operations.
- PLAY_MSG@(*A*) – Play back the message.
- READ_STATUS – poll status until CMD_FIN bit is set indicating play has finished.
- READ_INT – clear the interrupt.
- ERASE_MSG@(*A*) – erase the message.
- READ_STATUS – poll status until CMD_FIN bit is set indicating erase has finished.
- READ_INT – clear the interrupt.

11.3 Memory and Message Headers

The Memory and Message headers are located at the initial bytes of the 4kByte memory sector used to determine the format or function of the memory. The Memory Header stores users' configurable information including the memory protection scheme, the RMP and PMP pointers and the index table including POI, PU and other Voice Macros defined by the users. The Message Header located in both Reserved Sectors and Recording Memory stores the information for the device to determine what memory is available for recording and where the subsequent messages are stored.

11.3.1 Memory Header

Table 11-1 Memory Header

Initial Bytes of the Memory Header						
Byte0	**Byte1-2**	**Byte3-4**	**Bytes5-10**	**Bytes11-16**	**Byte17-22**	**Byte23-28**
0xCX	RMP[7:0]	PMP[7:0]	POI_VM	PU_VM	GPIO_WAKEUP	VM/VP[3]

The Memory Header contains at least seventeen bytes located at the beginning of the memory space. Byte0 determines whether or not the memory contains a Reserved Memory block as well as the memory protection scheme is used. Byte1-2 and Byte3-4 store the RMP and PMP pointers respectively. After the PMP, it is the start of the Voice Prompt/Voice Macro index table defined by the users. This table consists of six byte entries that are the start and end address of Voice Prompt or Voice Macro. Byte5-10, Byte11-15 & Byte 17-22 are reserved for the POI, PU and GPIO_WAKEUP Voice Macro which are the first three entries in this table (index 0, index 1 and index 2) to be executed on power-on initialization, power-up and GPIO_WAKEUP respectively. If this function is not desired, these entries should be left erased (0xFFFFFF,0xFFFFFF). When a PLAY_VP(i) or EXE_VM(i) command is sent to the ISD61S00, it reads the index table entry at address 6i+5 and executes the VP or VM at the address present in the table.

If the first byte Byte0 of the memory header is 0xFF, it means that no Reserved Memory is allocated, no memory protection is enabled and the whole memory will be available for record and playback only. To allocate the Reserve Memory, the bit4 of the Byte0 has to be set to zero. When Reserved Memory is allocated and memory protection is enabled upon users' definition, the both RMP (Recording Memory Pointer) and PMP (Protected Memory Pointer) will then created and stored at Byte1-2 and Byte3-4 of the Memory Header respectively. The ISD61S00 will not attempt to use memory before the RMP for

message recordings. This memory thus can be used for Voice Prompts, Voice Macros, User Data and Reserved Sectors. The PMP points to the boundary of protected memory and is used in conjunction with the RP, WP and CEP bits to set memory protection indicated below (also see Section 11.6 for details).

Table 11-2 The first byte of the Memory Header

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

11.3.2 Message Header

Table 11-3 Message Header

Message Header Byte							
Bit 7	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
BOM	EOM	RSVD	-	-	-	-	-

The ISD61S00 adopts a sector-based message management scheme that treats each 4kByte sector as a message frame. Each sector has a header indicating the state of the frame:

- BOM: “0” indicates that this sector is the beginning of a message.
- EOM: “0” indicates that this sector is the end of a message.
- RSVD: “0” indicates that there is a message recorded in this sector or this sector has been reserved for User Data.

Thus the number of the BOMs counted by scanning the Message Header of the Recording Memory will tell the number of the messages recorded in the ISD61S00.

When a sector is not an EOM sector, the next two bytes in the frame are the sector address pointing to the subsequent message. Thus the whole message can be re-composed by tracing sector address until an EOM is found.

It is not recommended to reserve the User Data inside the Recording Memory. However, for those instances where User Data needs to be allocated inside the Recording Memory, it is imperative that the RVSD has set and maintained at “0”. Otherwise, there is a risk that the User Data may be over-written by subsequent recordings.

11.4 Digital Access of Memory

ISD61S00 memory can be accessed as conventional digital memory using the DIG_READ and DIG_WRTIE SPI commands. This allows the user to:

- Reserve areas of memory for use as digital non-volatile memory as User Data
- Update Voice Prompts and macros (pre-recorded audio) in system.
- Read and verify Voice Prompt memory.
- Read sector headers of memory to determine memory usage.

The digital read and write commands can be issued even while an audio record or playback is in progress. The RDY/BSYB pin governs the flow control for all digital operations.

11.5 Device Erase Commands

ISD61S00 provides several ways to erase the flash memory. The flash memory has a minimum erasable sector size of 4kBytes. The sector erase command is sent with a start and stop sector address. The ISD61S00 also has commands to mass erase the memory.

11.6 Memory Contents Protection

Under certain circumstances, it is desirable for the users to protect portions of the internal memory from write/erase or interrogation (read). The ISD61S00 provides a method to achieve this by setting a protection memory pointer (PMP) that allows the users to protect internal memory for an address range from the beginning of memory to this sector where PMP is pointed. The type of protection is set by three bits in the memory header byte.

- The **CEP** (Chip Erase Protect) bit set to zero enables chip erase protection. This prevents a mass erase function, allowing the device to be configured as a write-once part. With the **CEP** bit set to one, even with write protection enabled, the part can be mass erased. After mass erasure, the initial sector byte defaults to no protection so the device can be re-programmed.
- The **WP** (Write Protect) bit set to zero enables write protection of the internal memory below the sector pointed to by the PMP. Write protection means that digital write or erase commands will not function in this memory area. This can be used to ensure that audio or data is not inadvertently erased or overwritten. The **WP** bit does not stop the execution of a REC_MSG@ or ERASE_MSG@ to messages with BOM headers in this memory.
- The **RP** (Read Protect) bit set to zero enables read protection of the internal memory below the sector pointed to by the PMP. Read protection means that digital read or audio playback commands through SPI or I^2S will not function in this memory area. This can be used to ensure that internal memory contents cannot be digitally copied or read.

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

12. DEVICE INITIALIZATION

Whenever the ISD61S00 detect as power-on reset condition or a high on the RESET pin of the device it begins a power-on initialization (POI) sequence. Whenever the ISD61S00 receives a power up command (PU) when it is in a power down state, it begins a power-up initialization (PU) sequence. Voice Macros VM(0) and VM(1) are reserved for POI and PU initialization routines. If no reserved memory exists or if the vectors VM(0) or VM(1) are not set, then a default routine is executed. The default sequence for POI is to power-down the ISD61S00. The default PU sequence is to power up and go to IDLE state.

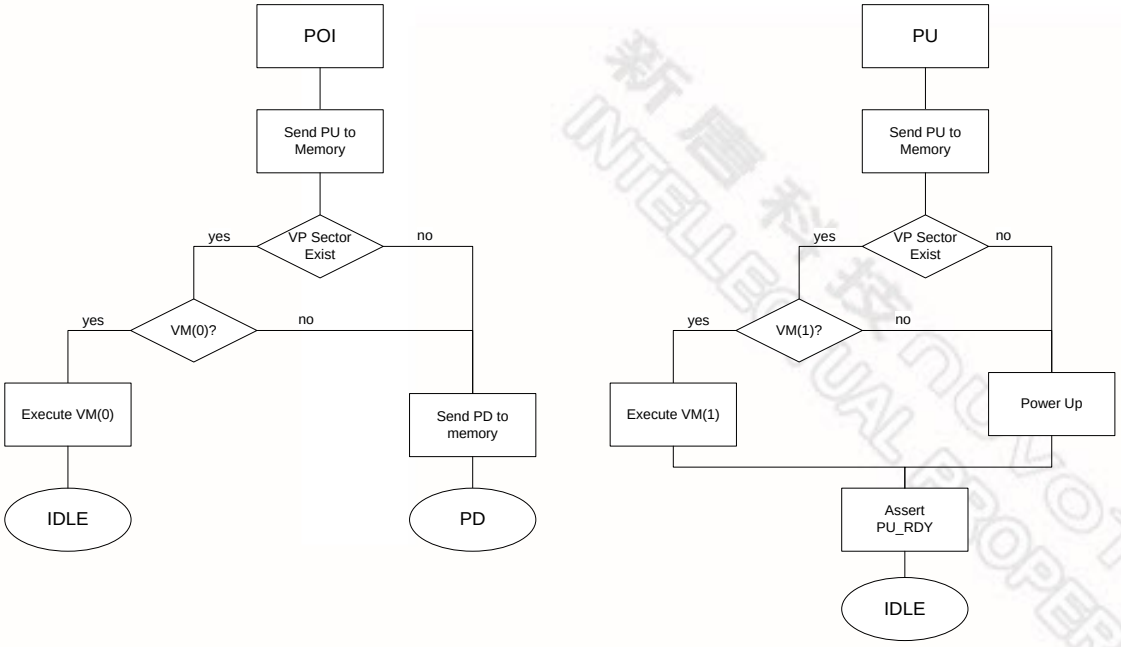


Figure 12-1 POI and PU Initialization Flowcharts

13. APPLICATION REFERENCE SCHEMATICS

The bottom part of Figure 13-1 shows a reference design for a PSTN interface to the ISD61S00. Two analog signal paths are provided for monitoring audio signals on TIP and RING. The Type 1 CID is for monitoring TIP and RING in an on-hook state to receive CID information. The Type II CID and speech interface is through a balanced configuration along with the output driver for use when the signal path is off-hook. Also included are optional circuits for line detection, ring detection and pulse dialing.

The Figure 13-2 illustrates a reference design for connecting the ISD61S00 to an integrated silicon DAA.

13.1 PCB Layout Guidelines

To gain maximum performance from the ISD61S00 it is important to provide clean analog supplies to the device. Separate Vcca and Vssa returns to the board low impedance point are essential for noise isolation. Good quality low ESR decoupliling capacitors on the supplies along with filter capacitors on Vbg and Vmid are also important for optimal performance. Care in shielding MIC connections from noise sources is imperative, these connections should be as short as possible and shielded with Vssa. The speaker and PO connections carry high currents and should be made as wide as possible. The TI inputs are differential and should be run as a pair and shielded with Vsaa. The flash SPI bus is a high speed serial bus and connections should be routed to the flash device in equal and short traces.

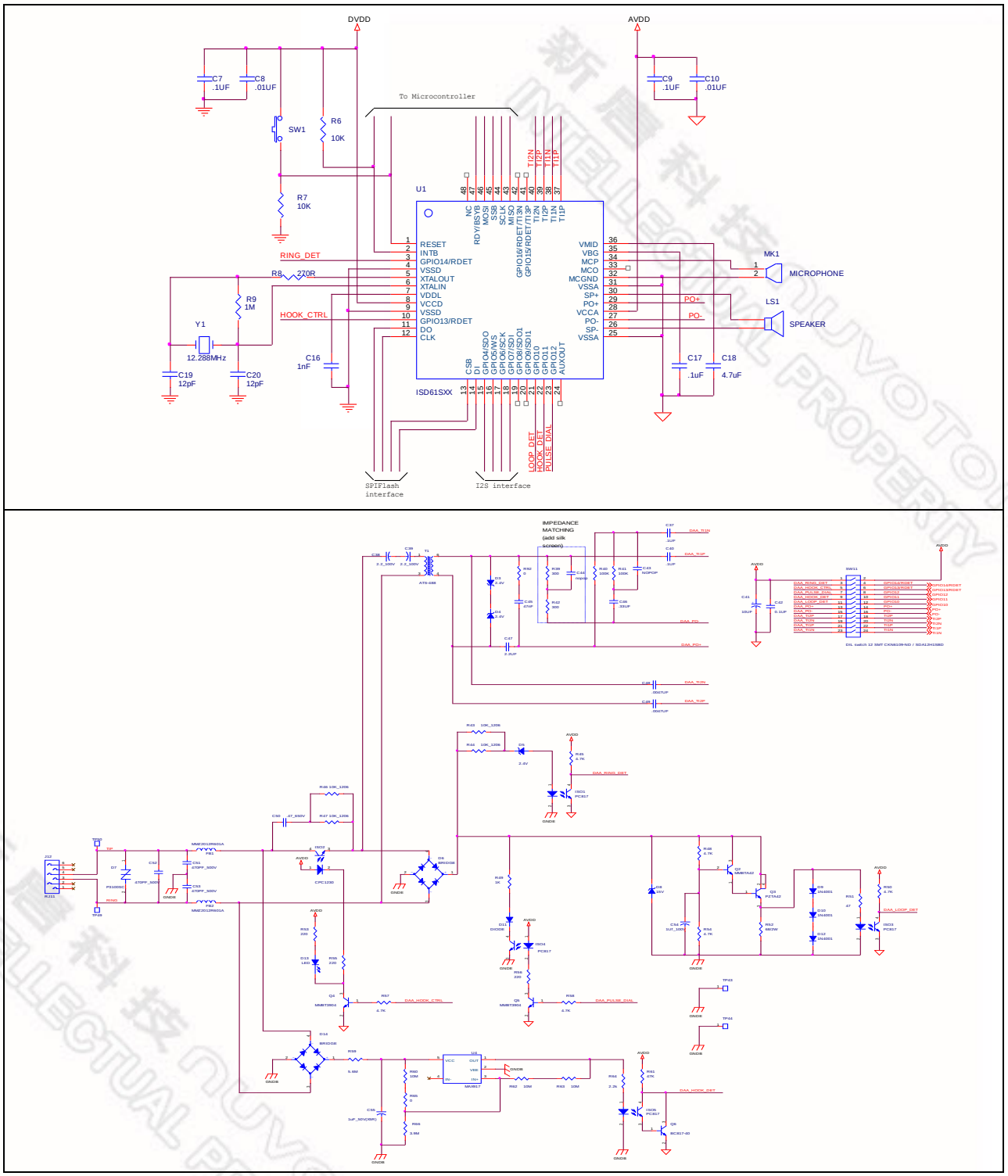
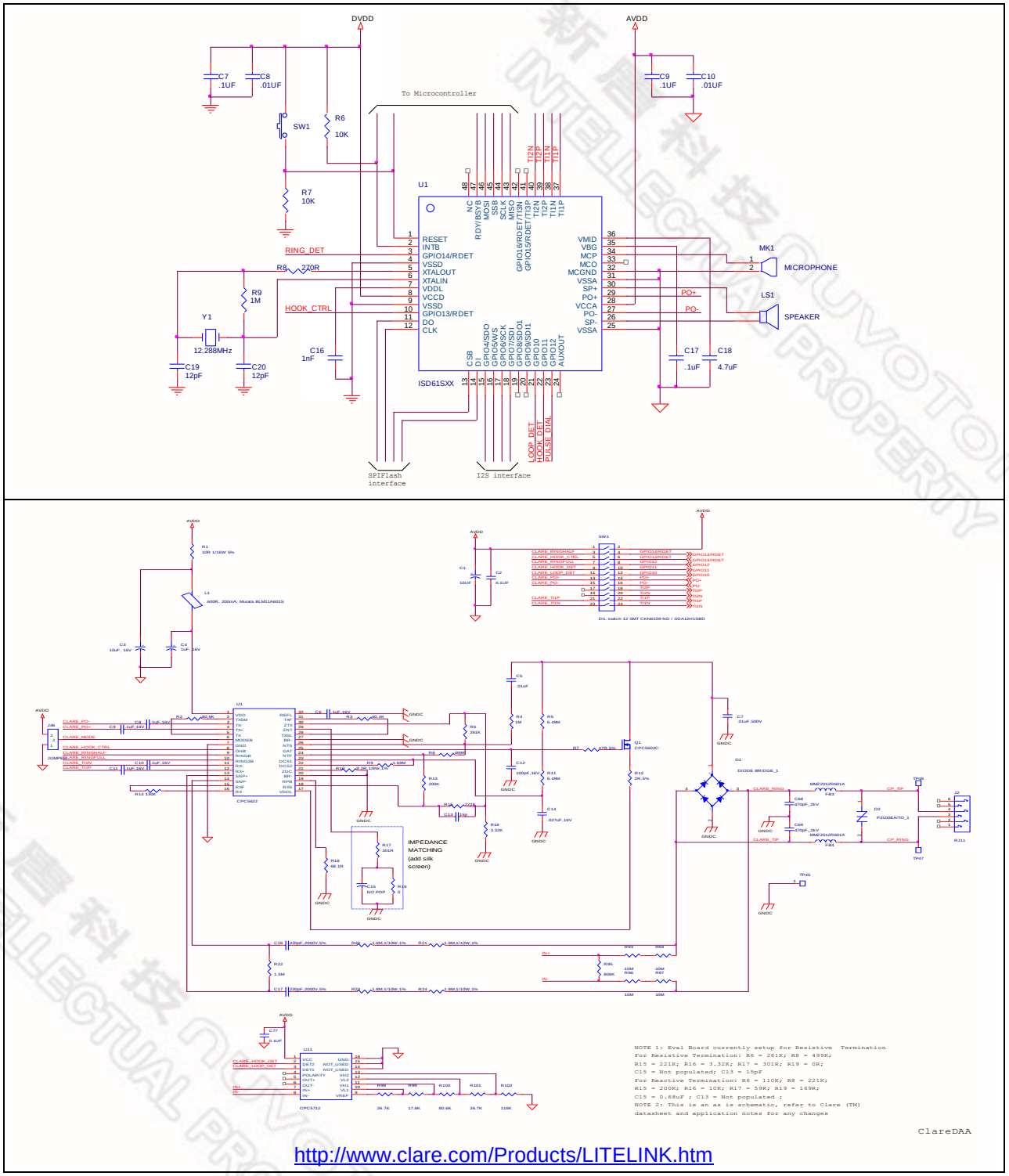


Figure 13-1 Application Reference Schematic with discrete DAA

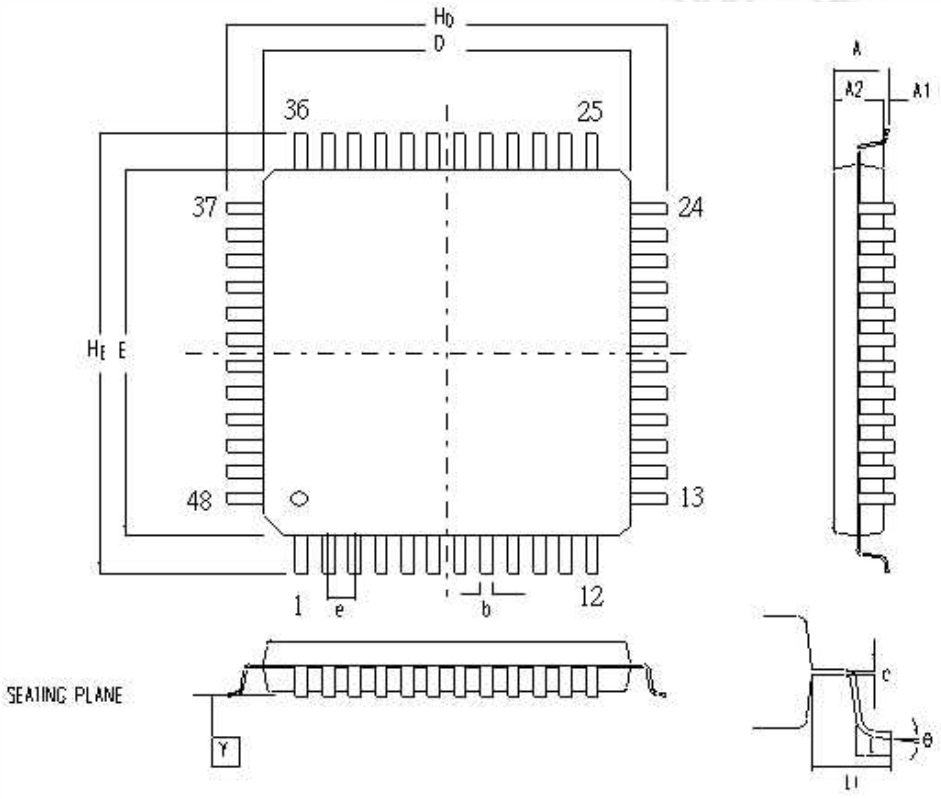


http://www.clare.com/Products/LITELINK.htm

Figure 13-2 Application Reference Schematic with integrated Clare DAA

14. PACKAGE SPECIFICATION

14.1 LQFP48L (7x7x1.4mm footprint 2.0mm)



Controlling dimension : Millimeters

Symbol	Dimension in inch			Dimension in mm		
	Min	Nom	Max	Min	Nom	Max
A	—	—	0.063	—	—	1.60
A1	0.002	0.004	0.006	0.05	0.10	0.15
A2	0.053	0.055	0.057	1.35	1.40	1.45
b	0.006	0.008	0.010	0.15	0.20	0.25
c	0.004	0.006	0.008	0.10	0.15	0.20
D	0.272	0.276	0.280	6.90	7.00	7.10
E	0.272	0.276	0.280	6.90	7.00	7.10
e	0.014	0.020	0.026	0.35	0.50	0.65
HD	0.350	0.354	0.358	8.90	9.00	9.10
HE	0.350	0.354	0.358	8.90	9.00	9.10
L	0.018	0.024	0.030	0.45	0.60	0.75
L1	—	0.039	—	—	1.00	—
Y	—	—	0.004	—	—	0.10
θ	0°	—	7°	0°	—	7°

15. ELECTRICAL CHARACTERISTICS

15.1 Absolute Maximum Ratings

Condition	Value
Junction temperature	150^0C
Storage temperature range	-65^0C to $+150^0$C
Lead temperature (soldering – 10 seconds)	300^0C
LQFP-48L Thermal Resistance, typical	76 C/W (ASE) 60C/W (Greatek) TBD
Voltage applied to any pin	(V_{SS} - 0.3V) to (V_{DD} + 0.3V)
Input current applied to any digital input pin	+/- 10 mA
ESD (Human Body Model)	2000 V
V_{DD} - V_{SS}	-0.5V to +3.63V
V_{DDL} - V_{SS}	-0.5V to + 1.98V
Device Power Dissipation	0.18 Watt (TBC)

- Stresses above those listed may cause permanent damage to the device. Exposure to the absolute maximum ratings may affect device reliability. Functional operation is not implied at these conditions.

15.2 Operating Conditions

OPERATING CONDITIONS (INDUSTRIAL PACKAGED PARTS)

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

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

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

15.3 DC Parameters

PARAMETER	SYMBOL	MIN	TYP [1]	MAX	UNITS	CONDITIONS
Supply Voltage	V_{DD}	2.7		3.6	V	
Input Low Voltage	V_{IL}	V_{SS}-0.3		0.3xV_{DD}	V	
Input High Voltage	V_{IH}	0.7xV_{DD}		V_{DD}	V	
Schmitt trig. Low to High threshold point	VT+	1.49	1.54	1.58	V	
Schmitt trig. high to low threshold point	VT-	1.24	1.29	1.34	V	
Pull-up resistor	R_{PU}	38	54	83	kΩ	
Pull-down resistor	R_{PD}	25	49	110	kΩ	
Output Low Voltage	V_{OL}			0.4	V	I_{OL} = 3μA
INTB Output Low Voltage	V_{OL1}			0.4	V	
Output High Voltage	V_{OH}	2.4			V	I_{OH} = -10μA
Internal logic supply	V_{DDL}	1.65	1.8	1.95	V	
Analog outputs DC level	V_{DC}		AV_{DD}/ 2		V	@ +27C
Half supply reference	V_{MIB}		1.65		V	AV_{DD}= 3.3V @ +27C
Bandgap reference	V_{BG}		1.218		V	AV_{DD}= 3.3V @ +27C
Operating Current (outputs loaded: 8 Ohm 120 Ohm)	I_{DD_MAX}		150		mA	AV_{DD}= 3.6V, Loaded, Sampling freq = 8 kHz, all blocks enabled, full scale.
Operating Current (outputs not loaded: 8 Ohm 120 Ohm)	I_{DD_MAX}		45		mA	AV_{DD}= 3.6V, No load, Sampling freq = 8 kHz, all blocks enabled, full scale.
Standby Current	I_{SB}		11	20	μA	AV_{DD}= 3.6V @ +27C
Input Leakage Current	I_{IL}			±10	μA	Force AV_{DD}

Notes: [1] Conditions V_{DD}=3.3V, T_A=25°C unless otherwise stated

15.4 Analog Transmission Characteristics

AVDD=3.3V ; V_{SS}=0V; T_A=+27°C; All ADC tests using Auxiliary input mode @ 0dB gain

PARAMETER	SYM	CONDITION	TYP	TRANSMIT (ADC)		RECEIVE (DAC)		UNIT
				MIN	MAX	MIN	MAX	
DC level	V_{DC}	DC level on the outputs SPP – SPN; POP - PON	AV_{DD}/2	--	--	1.35	1.8	V
Full Scale Level	T_{XMAX}	ADC (single ended) DAC (differential)	1.35 2.7	1.05 2.1	--- ---	1.05 2.1	--- ---	V_{PK} V_{PK}
Absolute Gain	G_{ABS}	-3dBFS @ 1020 Hz, AVDD =3.3V; T_A=+25°C;	0	-0.40	+0.40	-0.40	+0.40	dB
Absolute Gain variation with Supply Voltage	G_{ABSS}	AVDD=3.13V – 3.47V; -3dBFS @ 1020 Hz; T_A=+25°C	0	-0.50	+0.50	-0.50	+0.50	dB

15.5 Analog Distortion and Noise Parameters

All ADC tests using Auxiliary input mode @ 0dB gain

15.5.1 8kHz sampling

AVDD=3.3V; V_{SS}=0V; T_A=+27°C; 8kHz sampling,
high OSR selected (0x040[5]=1'b1),
dither turned off (0x151[0]=1'b0, 0x141[0]=1'b0)

PARAMETER	SYM	CONDITION	TRANSMIT (A/D)			RECEIVE (D/A)			UNIT
			MIN	TYP	MAX	MIN	TYP	MAX	
Signal to Noise Ratio	SNR	Idle channel A-weighted	80	90	--	80	90	--	dB
Total Harmonic Distortion	THD	-3dBFS @ 1020 Hz, 8Ohm speaker load A-weighted	--	-80	-75	--	-70	-60	dB
Frequency Response	F_{low}	-3dB Low pass cut-off		3.36			3.36		kHz
Power Supply Rejection	$PSRR_A$	V_{CCA}; 35mVrms DC to 3.4 kHz A-weighted	--	50	--	--	50	--	dB

15.5.2 16kHz sampling

AVDD=3.3V; V_{SS}=0V; T_A=+27°C; 8kHz sampling,
high OSR selected (0x040[5]=1'b1),
dither turned off (0x151[0]=1'b0, 0x141[0]=1'b0)

PARAMETER	SYM	CONDITION	TRANSMIT (A/D)			RECEIVE (D/A)			UNIT
			MIN	TYP	MAX	MIN	TYP	MAX	
Signal to Noise Ratio	SNR	Idle channel A-weighted	80	90	--	80	90	--	dB
Total Harmonic Distortion	THD	-3dBFS @ 1020 Hz, 8Ohm speaker load A-weighted	--	-80	-75	--	-70	-60	dB
Frequency Response	F_{low}	-3dB Low pass cut-off		6.73			6.73		kHz
Power Supply Rejection	$PSRR_A$	V_{CCA}; 35mVrms DC to 6.8 kHz A-weighted	--	50	--	--	50	--	dB

15.6 SPI Timing

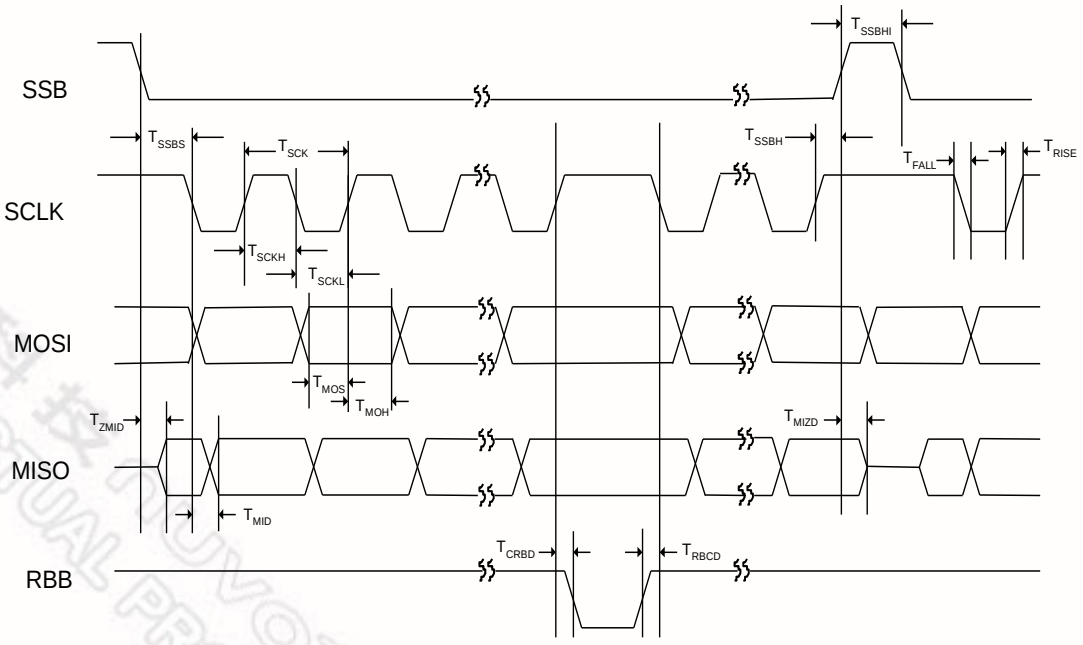


Figure 15-1 SPI Timing

SYMBOL	DESCRIPTION	MIN	TYP	MAX	UNIT
T_{SCK}	SCLK Cycle Time	100	---	---	ns
T_{SCKH}	SCLK High Pulse Width	45	---	---	ns
T_{SCKL}	SCLK Low Pulse Width	45	---	---	ns
T_{RISE}	Rise Time for All Digital Signals	---	---	10	ns
T_{FALL}	Fall Time for All Digital Signals	---	---	10	ns
T_{SSBS}	SSB Falling Edge to 1st SCLK Falling Edge Setup Time	60	---	---	ns
T_{SSBH}	Last SCLK Rising Edge to SSB Rising Edge Hold Time	30	---	---	ns
T_{SSBHI}	SSB High Time between SSB Lows	50	---	---	ns
T_{MOS}	MOSI to SCLK Rising Edge Setup Time	45	---	---	ns
T_{MOH}	SCLK Rising Edge to MOSI Hold Time	15	---	---	ns
T_{ZMID}	Delay Time from SSB Falling Edge to MISO Active	--	--	12	ns
T_{MIZD}	Delay Time from SSB Rising Edge to MISO Tri-state	--	--	12	ns
T_{MID}	Delay Time from SCLK Falling Edge to MISO	---	---	40	ns
T_{CRBD}	Delay Time from SCLK Rising Edge to RBB Falling Edge	--	--	12	ns
T_{RBCD}	Delay Time from RBB Rising Edge to SCLK Falling Edge	0	--	--	ns

15.7 Recommended Clock/Crystal Specification

The following crystal or external master clock specifications are recommended for a correct operation.

Parameter	Limit values			Unit	Condition
	Min.	Typ.	Max.		
Frequency	7.56	12.288	32.768	MHz	Fundamental mode
Load capacitance		18		pF	
Dynamic capacitance Cc		22.12		fF	
Resonance resistance Rc		40		Ω	
Electrostatic capacitance		5.1		pF	

15.8 Dual Tone Alert Signal (CAS)

AC Electrical Characteristics – Dual Tone Alert Signal Detection

DESCRIPTION	SYM.	MIN	TYP	MAX	UNITS	NO TEST
Low tone frequency	f_L		2130		Hz	
High tone frequency	f_H		2750		Hz	
Frequency deviation acceptation		± 1.0		± 3.0	%	1
Frequency deviation rejection		± 3.5			%	2
Accept signal level per tone		-40 -37.78		-2 0.22	dBV dBm	S/N=20 3, 5, 7
Reject signal level per tone				-46 -43.78	dBV dBm	S/N=20 4, 5, 7
Positive and negative Twist accept		-10		10	dB	S/N=20 6, 7
Signal to Noise Ratio		20			dB	5, 6, 7

NOTES:

1. The range within which tones are accepted.
2. The range outside of which tones are rejected.
3. This applies BT specification that has covered the requirements of Bell core.
4. This applies MITEL MT8843 specification. Winbond W91030A: -44 dBm, newave NW6006: -47 dBV and CLI CMX602A: -46 dBV.
5. These characteristics are for AVDD=3.3V and 25°C.
6. Both tones have the same amplitude and at the nominal frequencies.

$$\text{Twist} = 20\log\left(\frac{f_H \text{amplitude}}{f_L \text{amplitude}}\right)$$

6. Band limited random noise 300~3400 Hz. Measurement valid only when the tone is present.

AC Timing Characteristics – Dual Tone Alert Signal Detection

Symbol	DESCRIPTION	MIN	TYP	MAX	UNITS	NOTES
t_{CASDP}	Alert signal present detect time	0.5	1.88	10	ms	1
t_{CASDA}	Alert signal absent detect time	0.5	1.46	10	ms	2

NOTES:

1. t_{CASDP} Typical time corresponding to 4 cycles of low tone
2. t_{CASDA} Typical time corresponding to 4 cycles of high tone

15.9 FSK Detection – 1200baud Bell 202, ITU V.23, 300 baud Bell 103, ITU V.21

AC Electrical Characteristics – FSK Detection

DESCRIPTION	SYM.	MIN	TYP	MAX	UNITS	NOTES
Input frequency detection						
Bell 202 '1' (Mark)	f_{MARK}	1188	1200	1212	Hz	
Bell 202 '0' (Space)	f_{SPACE}	2178	2200	2222		
ITU-T V.23 '1' (Mark)		1280.5	1300	1319.5		
ITU-T V.23 '0' (Space)		2068.5	2100	2131.5		
Transmission Rate			1200		Baud	± 1%
			150		Baud	± 1%
			75		Baud	± 1%
Input detection level per tone		-40 -37.78		-6.2 -4.0	dBV dBm	S/N=20 1,3
Reject signal level per tone				-48 -45.78	dBV dBm	S/N=20 1,3
Positive and negative Twist accept		-10		10	dB	S/N=20 2,3
Signal to Noise Ratio		20			dB	S/N=20 3

DESCRIPTION	SYM.	MIN	TYP	MAX	UNITS	NOTES
Input frequency detection						
Bell 103 '1' (Mark), high band	f_{MARK}	2213	2225	2237	Hz	
Bell 103 '0' (Space), high band	f_{SPACE}	2013	2025	2037	Hz	
Bell 103 '1' (Mark), low band	f_{MARK}	1258	1270	1282	Hz	
Bell 103 '0' (Space), low band	f_{SPACE}	1058	1070	1082	Hz	
ITU-T V.21 '1' (Mark), high band	f_{MARK}	1638	1650	1662	Hz	
ITU-T V.21 '0' (Space), high band	f_{SPACE}	1838	1850	1862	Hz	
ITU-T V.21 '1' (Mark), low band	f_{MARK}	968	980	992	Hz	
ITU-T V.21 '0' (Space), low band	f_{SPACE}	1168	1180	1192	Hz	
Transmission Rate			300		Baud	± 1%
			110		Baud	± 1%
Input detection level per tone		-40 -37.78		-6.2 -4.0	dBV dBm	S/N=20 1,3
Reject signal level per tone				-48 -45.78	dBV dBm	S/N=20 1,3
Positive and negative Twist accept		-10		10	dB	S/N=20

						2,3
Signal to Noise Ratio		20			dB	S/N=20 3

NOTES:

1. These characteristics are for AVDD=+3.3V and 25°C.
2. Both mark and space have the same amplitude and at the nominal frequencies.

$$\text{Twist} = 20\log\left(\frac{\text{amplitude of } f_{MARK}}{\text{amplitude of } f_{SPACE}}\right)$$

3. Band limited random noise (200~3400 Hz). Measurement is valid only when the FSK signal is present. Note that the BT band is 300~3400 Hz, while the Bellcore band is 0~4K Hz.

新唐科技 NUVOTON INTELLECTUAL PROPERTY

15.10 FSK Transmitter – Bell 202, ITU-V.23, Bell 103, ITU-V.21

Modulation rates and characteristic frequencies for the forward data-transmission channel

DESCRIPTION	SYM	MIN	TYP	MAX	UNITS	NOTES
Twist		-0.5	0	0.5	%	
Baud Rate			1200		Baud	± 1%
Bell 202						
"1" (Mark)	f_{MARK}	1197	1200	1203		$\pm 3Hz$
"0" (Space)	f_{SPACE}	2197	2200	2203		$\pm 3Hz$
V.23						
"1" (Mark)	f_{MARK}	1297	1300	1303		$\pm 3Hz$
"0" (Space)	f_{SPACE}	2097	2100	2103		$\pm 3Hz$

DESCRIPTION	SYM	MIN	TYP	MAX	UNITS	NOTES
Twist		-0.5	0	0.5	%	
Baud Rate			300		Baud	± 1%
			110		Baud	± 1%
Bell 103						
"1" (Mark), high band	f_{MARK}	2222	2225	2228		$\pm 3Hz$
"0" (Space), high band	f_{SPACE}	2022	2025	2028		$\pm 3Hz$
"1" (Mark), high band	f_{MARK}	1267	1270	1273		$\pm 3Hz$
"0" (Space), high band	f_{SPACE}	1067	1070	1073		$\pm 3Hz$
V.21						
"1" (Mark), high band	f_{MARK}	1647	1650	1653		$\pm 3Hz$
"0" (Space), high band	f_{SPACE}	1847	1850	1853		$\pm 3Hz$
"1" (Mark), low band	f_{MARK}	977	980	983		$\pm 3Hz$
"0" (Space), low band	f_{SPACE}	1177	1180	1183		$\pm 3Hz$

*Tx signal % baud or bit rate accuracy is the same as XTAL/CLOCK % frequency accuracy.

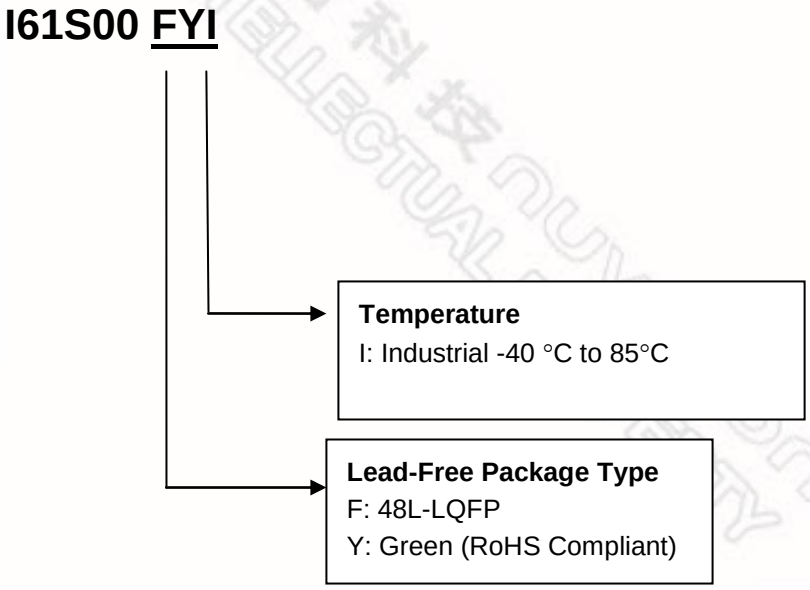
15.11 DTMF Detection

AC Electrical Characteristics – DTMF detection

Description	Sym	Min	Typ	Max	Units	Notes
Frequency Deviation Acceptation				±2.0	%	
Frequency Deviation Rejection		±3.5			%	
Accept signal level per tone		-36		-6	dBm	
Reject signal per tone				-46	dBm	
Positive and negative Twist accept				10	dB	S/N=20
Signal to Noise Ratio		20			dB	

新唐科技 NUVOTON INTELLECTUAL PROPERTY

16. ORDERING INFORMATION



新唐科技 NUVOTON INTELLECTUAL PROPERTY

17. REVISION HISTORY

VERSION	DATE	PAGE	DESCRIPTION
2.00	November 28, 2008		• Release as Design Guide.
2.01	October 2009 – not yet released		• Update AGC parameters. • Update AGC description. • Description in READ_STATUS command, number of status bytes.
2.5	May 5th, 2010		• Update pin-out diagram. • Update block diagram.
2.6	Oct 22nd, 2010		• Add table of supported serial flash memory.
2.7	Mar 10, 2011		• Update SPI timing.

Important Notice

Nuvoton products are not designed, intended, authorized or warranted for use as components in systems or equipment intended for surgical implantation, atomic energy control instruments, airplane or spaceship instruments, transportation instruments, traffic signal instruments, combustion control instruments, or for other applications intended to support or sustain life. Furthermore, Nuvoton products are not intended for applications wherein failure of Nuvoton products could result or lead to a situation wherein personal injury, death or severe property or environmental damage could occur.

Nuvoton customers using or selling these products for use in such applications do so at their own risk and agree to fully indemnify Nuvoton for any damages resulting from such improper use or sales.

Please note that all data and specifications are subject to change without notice.
All the trademarks of products and companies mentioned in this datasheet belong to their respective owners.

Figure 3-1 ISD61S00 48-Lead LQFP Pin Configuration 12
Figure 8-1 SPI Data Transaction 35
Figure 8-2 R/B Timing for SPI Writing Transactions 35
Figure 8-3 SPI Transaction Ignoring RDY/BSYB 36
Figure 8-4 Clock Generation Block Diagram 41
Figure 8-5 PMC/I2S signal routing 47
Figure 8-6 Microphone Voltage Gain Mode 57
Figure 8-7 Microphone Current Gain Mode 58
Figure 8-8 Current Mode Gain with Electret MIC model 59
Figure 8-9 Auxiliary Input Mode 60
Figure 8-10 TI Input Amplifier 61
Figure 8-11 Analog Output 63
Figure 8-12 DTMF Detector - Functional Block Diagram 65
Figure 8-13 DTMF Detector Acquisition Timing 66
Figure 8-15 Tone Generator Block Diagram 72
Figure 8-16 Architecture of Linear FSK Waveform Generator 77
Figure 8-17 Block Diagram of FSK Detector 86
Figure 8-18 band pass filter frequency response 95
Figure 8-19 Block Diagram of CAS Detection 100
Figure 8-20 Detector Biquad Low Tone Frequency Response 100
Figure 8-21 CAS Detector Biquad High Tone Frequency Response 101
Figure 8-22 CAS Internal IIR Filter Frequency Response Figure 103
Figure 8-23 FSK ATD IIR filter structure 107
Figure 8-24 FSK encoder IIR filter structure 108
Figure 8-25 Band Pass filter frequency response of voice energy detection 115
Figure 8-26 Call progress tone band-pass filter frequency response 117
Figure 8-27 Block diagram of ring detector 120
Figure 8-28 The timing of ring detector with RNG_CTRL[1:0]=2'b11 120
Figure 8-29 Gain/Mixer Channel 125
Figure 8-30 CODEC/EC Gain Stages 125
Figure 8-31 Gain Stages and Mixer with Control Addresses 128
Figure 8-33 Ringer Tone Generator Block 137
Figure 9-1: Signal flow through the Acoustic Echo Canceller (AEC) in the speech Processor. 140
Figure 9-2: Signal flow through the Line Echo Canceller (LEC) in the speech Processor. 140
Figure 9-3 AGC Operation regions. 161
Figure 10-1 SPI Record 177

Figure 10-2 SPI Feed-through 177
Figure 10-3 SPI Playback 178
Figure 10-4 Feed-through SPI 179
Figure 10-5 SPI Send Compressed Data to Decode 181
Figure 10-6 SPI Received Encoded Data 182
Figure 11-1 ISD61S00 Memory Format 190
Figure 12-1 POI and PU Initialization Flowcharts 196
Figure 13-1 Application Reference Schematic with discrete DAA 197
Figure 13-2 Application Reference Schematic with integrated Clare DAA 198
Figure 15-1 SPI Timing 203

新唐科技 NUVOTON INTELLECTUAL PROPERTY

Table 7-1 Status Register Description 32
Table 7-2 Interrupt Status Register Description 33
Table 7-3 Interrupt Status Register Description 33
Table 7-4 Timer interrupt and operation status 34
Table 8-1 COMP_CFG Register Compression Type. 37
Table 8-2 CFG0 Sample Rate Control 37
Table 8-3: PLL Frequency Examples 42
Table 8-4 Microphone and Auxiliary mode settings 54
Table 8-5 Microphone Gain settings 55
Table 8-6 Microphone Bias settings 55
Table 8-7 Microphone Resistor settings 56
Table 8-8 TI Gain Settings 63
Table 8-9 TI MUX Settings 63
Table 8-11 DTMF Frequency Deviation Register 67
Table 8-12 Tone Generation Mode Setup 72
Table 8-13 DTMF Frequency Mapping. 76
Table 8-14 FSK Encoder Specification Selection 78
Table 8-15 FSK Detector Modes. 88
Table 8-16 FSK_CMP_CNT Definition. 90
Table 8-17 CAS Detector Frequency Deviation Control. 102
Table 8-18 CAS/ADT Detection Bits 104
Table 8-21 Mixer Gain Registers. 127
Table 9-1 AEC/LEC Control Register Map 141
Table 9-2 SC Gain Table 157
Table 10-1 SPI Commands 166
Table 10-2 Commands vs. Status 168
Table 11-1 Memory Header 193
Table 11-2 The first byte of the Memory Header 194
Table 11-3 Message Header 194