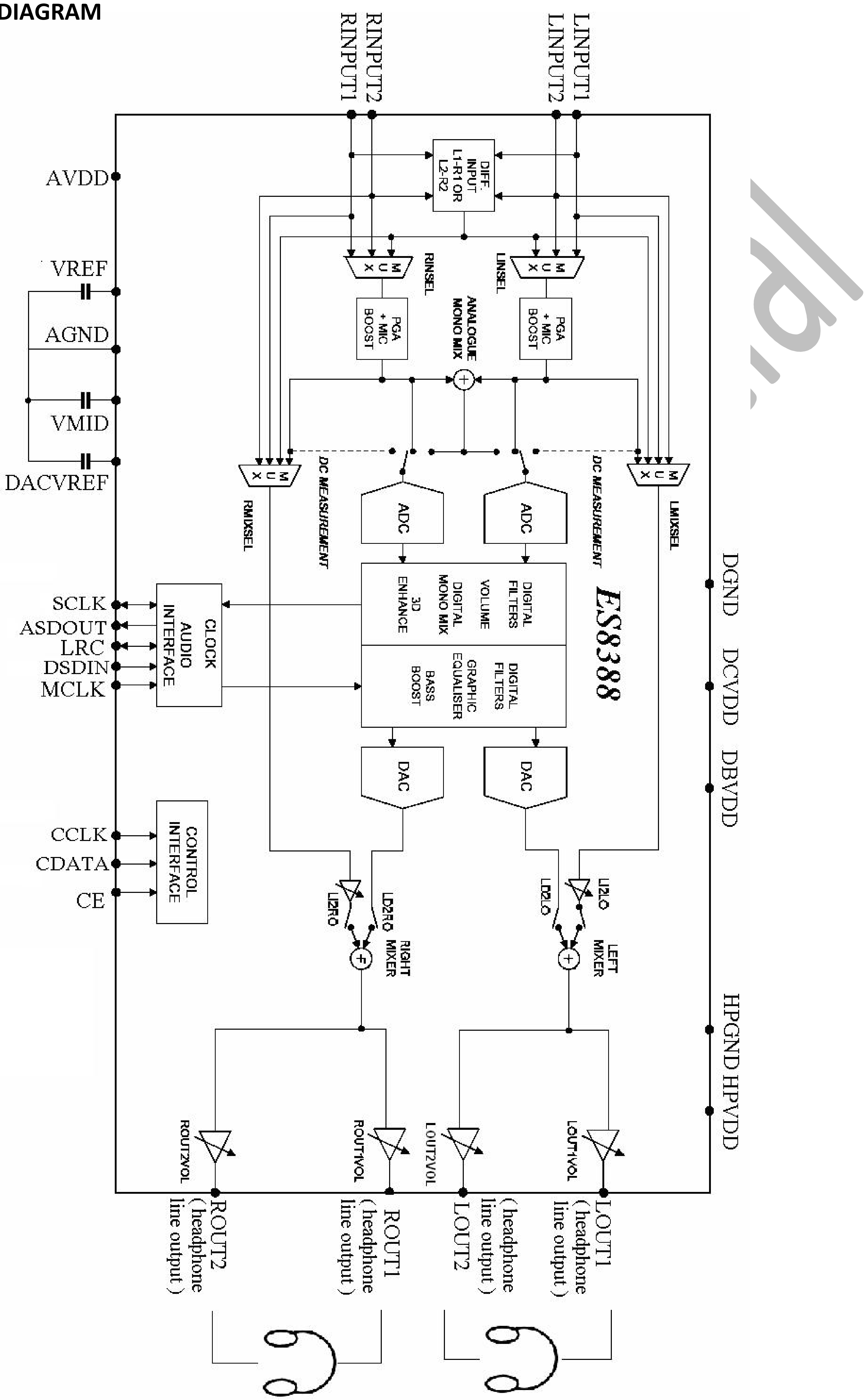
ES8388 User Guide

ES8388 User Guide

1 ES8388 BLOCK DIAGRAM



INPUT SIGNAL PATH

The two analogue inputs LINPUT1/2 RINPUT1/2 can be selected by a switch, and then followed by a PGA gain boost. The inputs can be individually selected or a differential input of either (LINPUT1 RINPUT1) or (LINPUT2 RINPUT2) may also be selected. These Inputs can be configured as microphone or line level. The signal then enters a high quality ADC. Alternatively, the two channels can also be mixed in the analogue domain and digitized in one ADC while the other ADC is turned off.

One on-chip ALC module can be used to control the signal level during recording. The gain of the PGA can be controlled either by the user or by the on-chip ALC function.

OUTPUT SIGNAL PATH

The output signal paths consist of digital filters, DACs, analogue mixers and output drivers. The digital filters and DACs are enabled in the “playback” or “record and playback” mode. In “bypass” mode, the analogue mixing and amplification can be utilized while DAC is disabled.

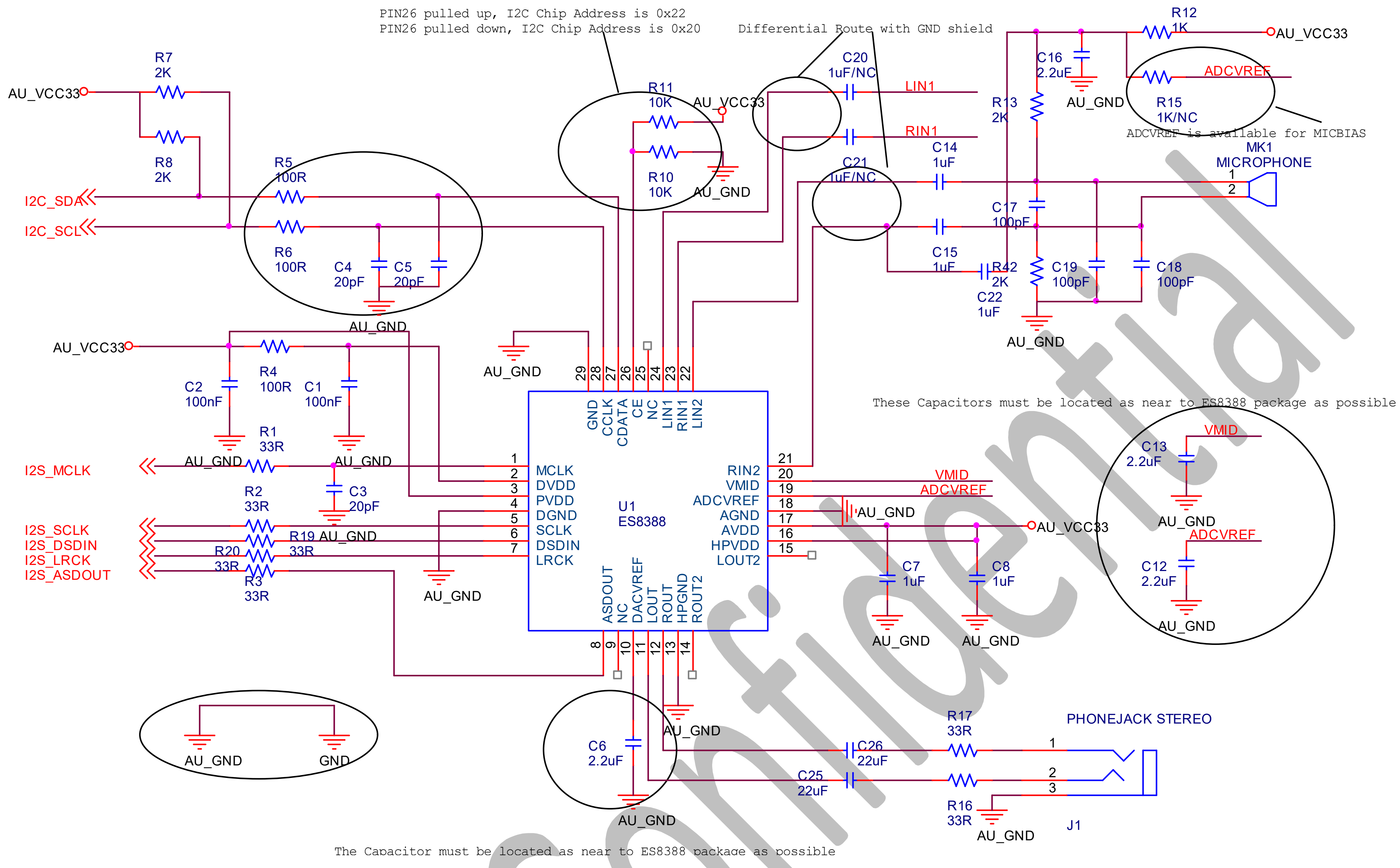
DACs output can be mixed with analogue signals from the input pins Linput1/2 and Rinput1/2, and the mixed signal is fed to the output drivers LOUT1/ROUT1, LOUT2/ROUT2. The mixers and output drivers can be separately enabled by individual control bits.

The LOUT1/ROUT1 and LOUT2/ROUT2 can drive a 16Ω (up to 40mW) or 32Ω stereo headphone or stereo line output.

2 RECOMMENDED OPERATING CONDITION

Digital supply range (Core)	**DVDD**	1.8V to 3.3V ***(If LRCK is lower than16KHZ, 1.8V DVDD power supply must be used to avoid noise)*** (Lowest power consumption at 1.8V)
Digital supply range (Buffer)	**PVDD**	1.8V to 3.3V
Analog supply range	**HPVDD, AVDD**	1.8V to 3.3V (Best audio performance at 3.3V)
Ground	**DGND, AGND, HPGND**	0V

3 TYPICAL APPLICATION CIRCUIT



Notes:

*** One LDO is recommended to provide power supply to ES8388 because ES8388 is an analog device which will be sensitive to noise.**

*** If LRCK frequency is lower than 16KHZ, 1.8V DVDD power supply must be used to avoid noise.**

*** A differential Microphone circuit is recommended to ES8388.**

***The signal MIC_INP and MIC_INN must be parallel with each other on PCB layout.**

*** Filter and decoupling capacitors should be located as close to ES8388 package as possible during layout.**

*** Two 1K pull up resisters are recommended to I2C bus. R-C low pass filter is recommended to I2C CLK.**

*** If PIN26 (CE) be pulled down to GND, I2C Chip ID is 0x20, otherwise 0x22.**

4 CIRCUIT AND PCB LAYOUT GUIDE

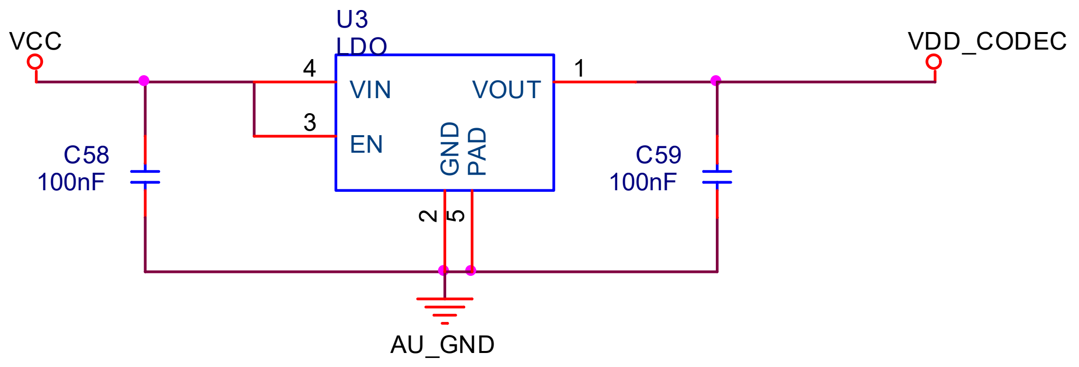
Audio Codec, such as ES8388, should be used as analog IC. They are sensitive to noise. So during circuit design and PCB layout, there are some issues which should be careful.

- Use a LDO to power supply for audio codec if it is possible

Audio Codec is usually sensitive to noise. In general, codec must have one quiet power supply which is isolated from system power supply.

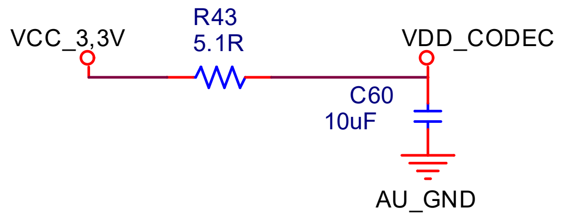
To get the best audio performance, Audio codec require a quiet power supply. It is recommended that one LDO is dedicated to Audio Codec.

Example 1 for Audio codec power supply,



In practical application, Audio Codec may share power supply with CPU or Memory. In this case, if LDO can not be used due to BOM cost or the other reason, a RC filter should be recommended for audio codec power supply to isolate Audio Codec from peripheral circuit.

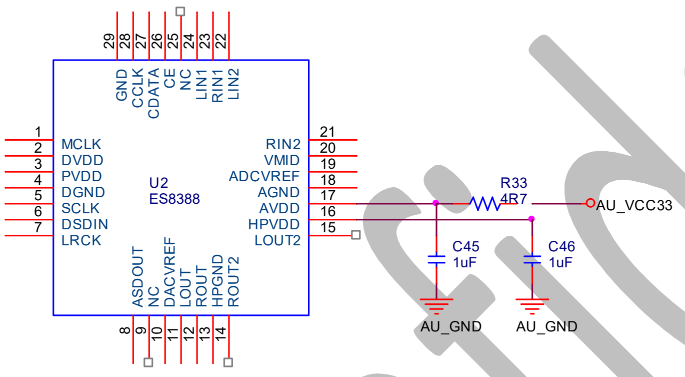
Example 2 for Audio Codec power supply



- ➢ Use a small resistor to isolate AVDD from HPVDD.

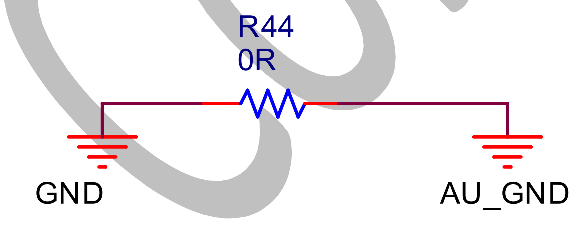
In practical application, please use a 1ohms or 4.7ohms resistor to isolate AVDD from HPVDD. Better Audio performance, such as THD+N, SNR and Crosstalk, will be get if this resistor is used.

The following drawing is an example for isolation on AVDD and HPVDD. Here, R33 is the resistor which is used to isolate the AVDD from HPVDD.



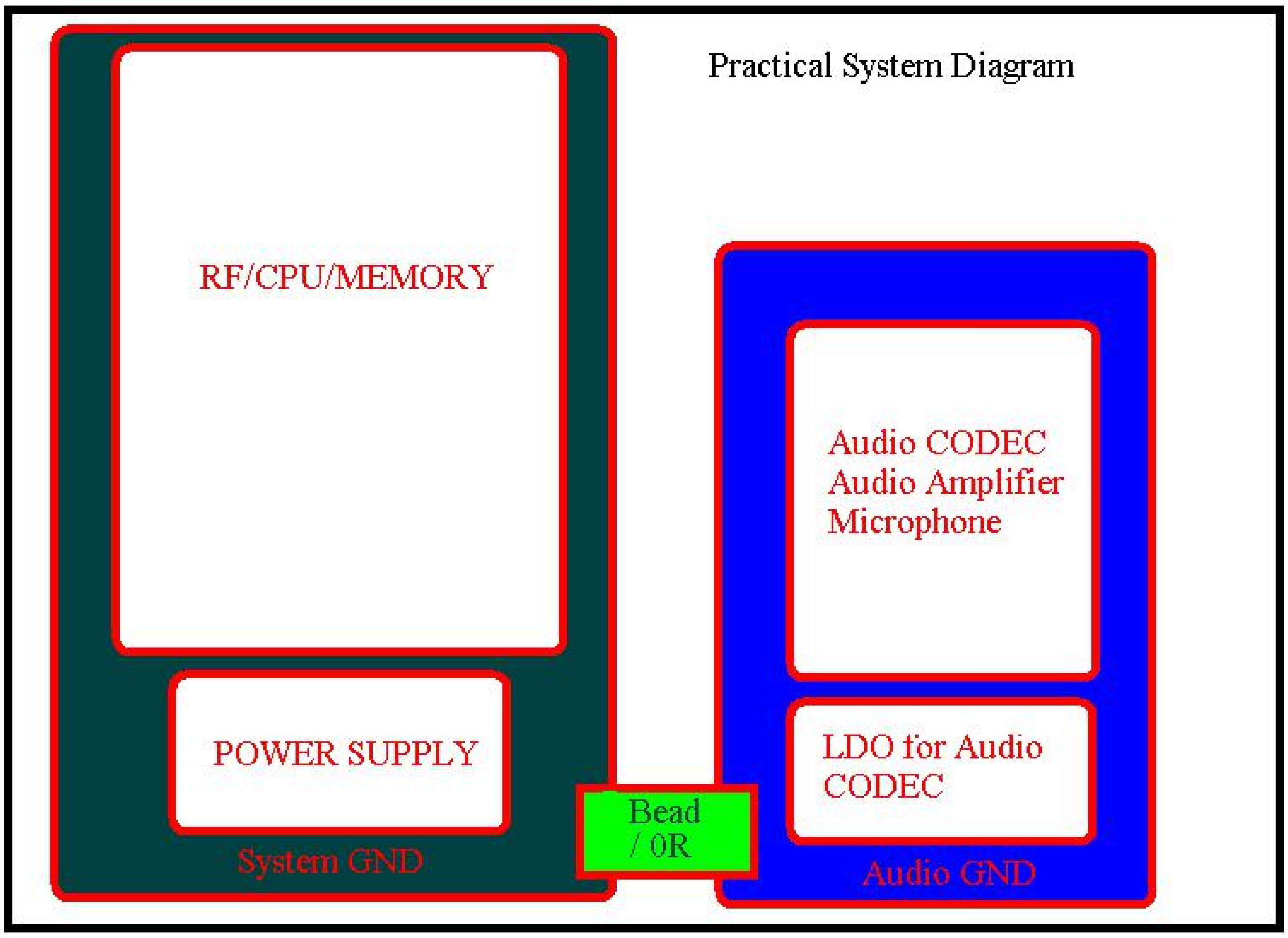
- ➢ Separate Audio Ground from System Ground

Please use a magnetic bead or small resistor to separate audio Ground from System Ground, especially in RF system. The following drawing is an example for GND isolation. Here is used to isolate Audio GND (Net: AGND) from system GND (Net: GND).



✧ PCB Layout highlight

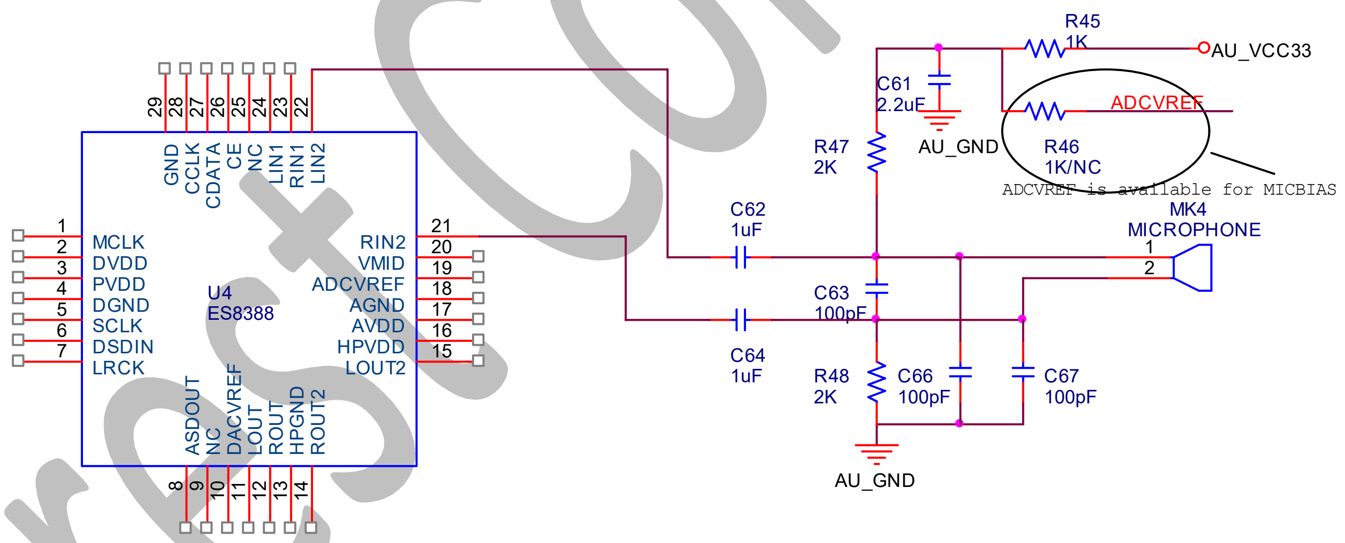
Please pay more attention to GND layout to get better audio performance. CODEC is usually sensitive to noise. Please let CODEC has one separate GND path if it is possible. Don’t let CODEC’s GND return current share the same path with others, especially in RF system.



- Microphone circuit

It is recommended that differential Microphone circuit is used for audio codec. The differential mode can get better Microphone performance, such as larger microphone volume and higher microphone SNR.

The following is the example for Microphone circuit.



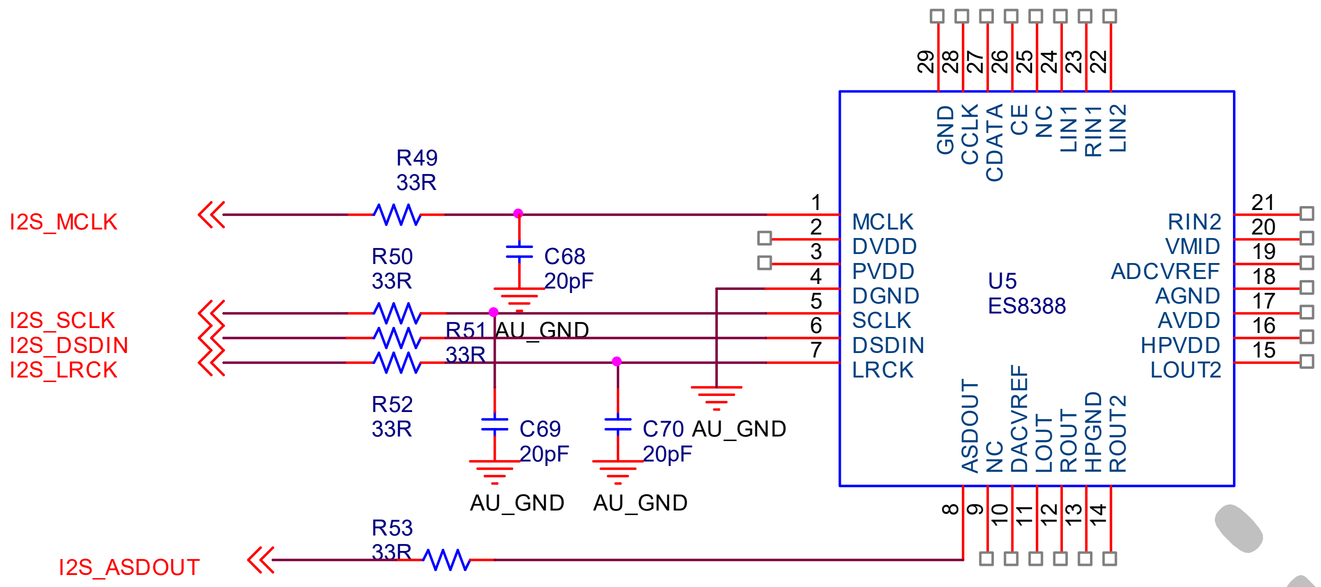
Please note that ES8388 can’t provide Microphone Bias voltage. So In the above microphone differential input circuit, AVDD is used for microphone bias voltage , a RC filter(R45 and C61) is used as LPF filter to attenuate noise on AVDD.

The ADCVREF of ES8388 also can be used as Microphone Bias voltage for electret microphone or MEMS microphone.

✧ PCB Layout highlight

The microphone signal is very sensitivity for noise. It is recommended that the microphone signal path is shielded by GND path, especially where there is long microphone signal path.

- I2S signal



5 I2C / SPI INTERFACE

I2C is a bi-directional bus which can be used to read or write register. SPI bus should only be used to write register.

- **I2C BUS**

Don’t connect CE pin to IO of MCU, CPU or DSP. CE pin should be pulled up to PVDD or pulled down to DGND.

The chip address for I2C is 0x20 if CE pin is pulled down to DGND. The chip address for I2C is 0x22 if CE pin is pulled up to PVDD.

There are two pull-up resisters on I2C CCLK pin and I2C CDATA pin. 1KΩ resister is recommended for the pull-up resisters.

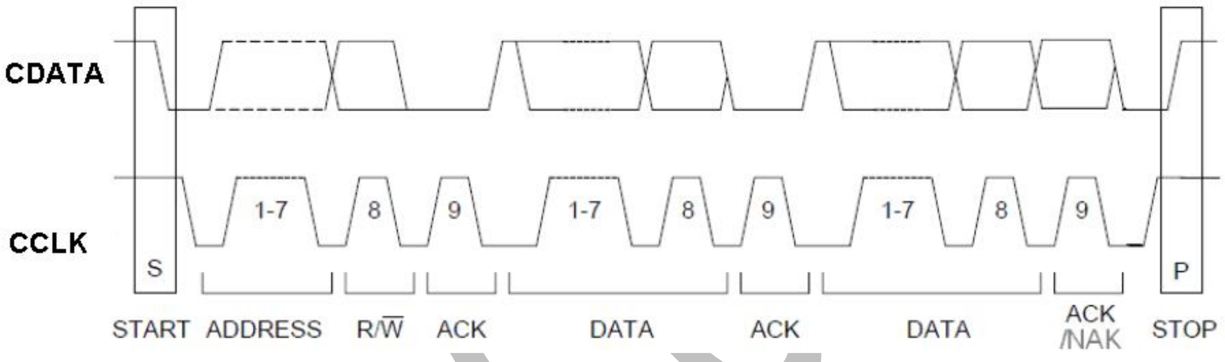


Figure 2. I2C timing

	Chip Address		R/W		Register Address		Data to be written		
Start	001000	AD0	0	ACK	RAM	ACK	DATA	ACK	Stop

Figure 3. I2C write

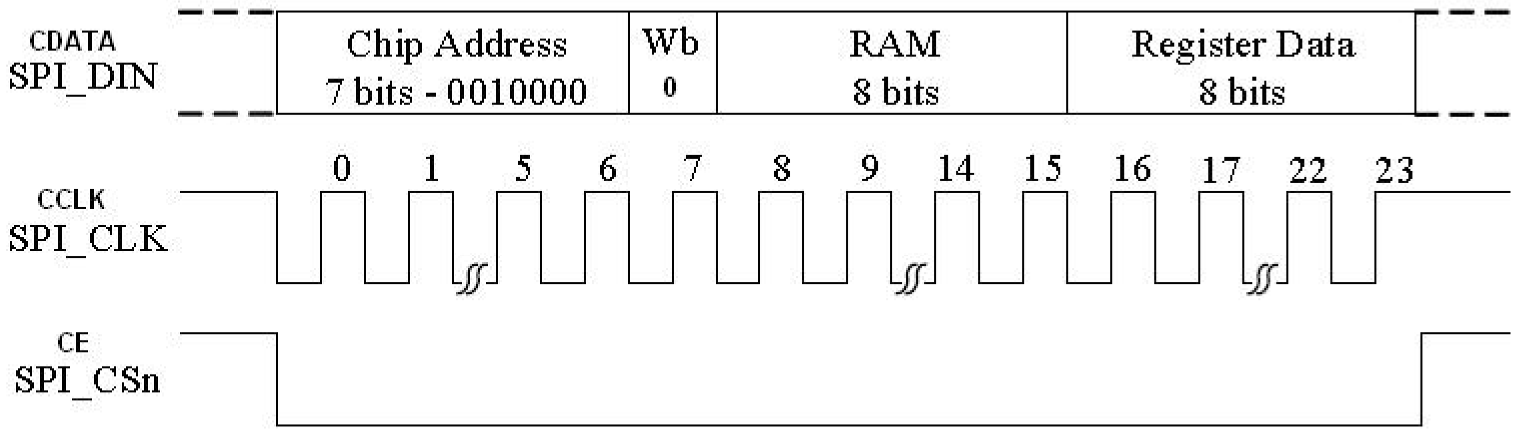
	Chip Address		R/W		Register Address		
Start	001000	AD0	0	ACK	RAM	ACK	
	Chip Address		R/W		Data to be read		
Start	001000	AD0	1	ACK	DATA	NAK	Stop

Figure 4. I2C Read

- **SPI BUS**

CE pin should be connected to IO of MCU, CPU or DSP. Don’t connect CE pin to DGND. The chip address for SPI is 0x20.

SPI bus should only be used to write register.



RAM = Register Address Mapping

6 DIGITAL AUDIO INTERFACE

The digital audio interface is used for inputting DAC data to ES8388 and outputting ADC data from it. The digital audio interface uses four pins:

- **ASDOUT :** ADC data output
- **DSDIN** : DAC data input
- **LRCK** : Left/Right data alignment clock
- **SCLK** : Bit clock

The clock signals SCLK and LRCK can be outputs when ES8388 operates as a master, or inputs when it is a slave.

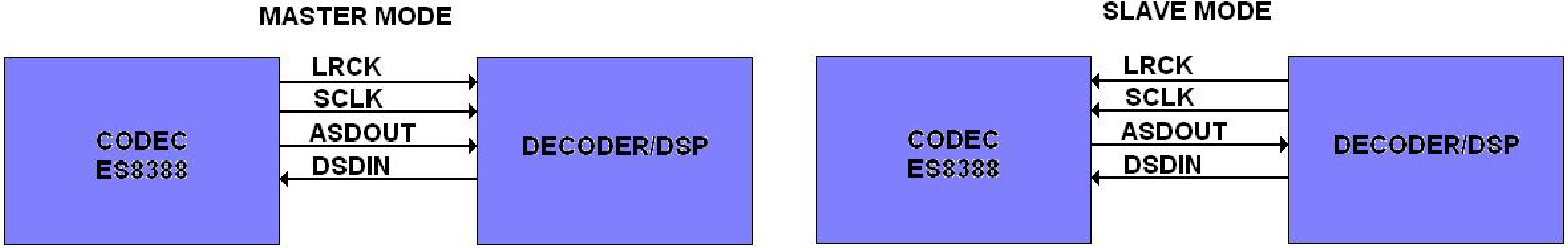
ES8388 can support four different audio data formats:

- **I2S**
- **Left Justified**
- **Right Justified**
- **DSP mode (PCM)**

All four of these formats are MSB first.

6.1 MASTER AND SLAVE MODE OPREATION

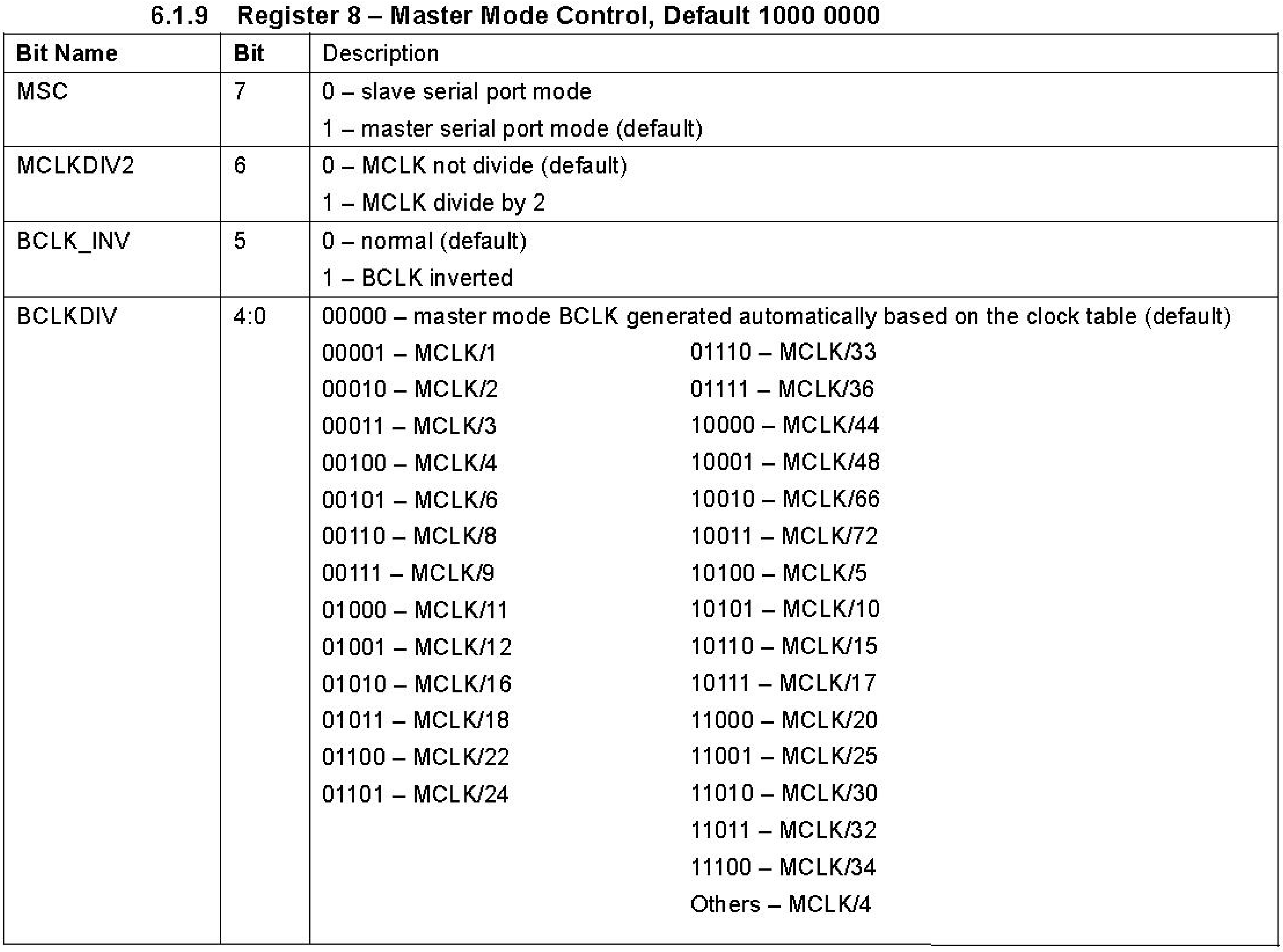
ES8388 digital audio interface can operates as a master or a slave as shown in the figure listed below.



In master mode, SCLK is derived from MCLK via a programmable division set by BCLK_DIV. LRCK is derived from MCLK via a programmable division set by ADCFsRatio or DACFsRatio

In slave mode, ES8388 can auto check MCLK/LRCK ratio and MCLK/SCLK ratio.

The Bit7(MSC) in register 0x08 should be used to set ES8388 in Master or Slave mode.



6.1.9 Register 8 – Master Mode Control, Default 1000 0000

Bit Name	Bit	Description
MSC	7	0 – slave serial port mode 1 – master serial port mode (default)
MCLKDIV2	6	0 – MCLK not divide (default) 1 – MCLK divide by 2
BCLK_INV	5	0 – normal (default) 1 – BCLK inverted
BCLKDIV	4:0	00000 – master mode BCLK generated automatically based on the clock table (default) 00001 – MCLK/1 00010 – MCLK/2 00011 – MCLK/3 00100 – MCLK/4 00101 – MCLK/6 00110 – MCLK/8 00111 – MCLK/9 01000 – MCLK/11 01001 – MCLK/12 01010 – MCLK/16 01011 – MCLK/18 01100 – MCLK/22 01101 – MCLK/24 01110 – MCLK/33 01111 – MCLK/36 10000 – MCLK/44 10001 – MCLK/48 10010 – MCLK/66 10011 – MCLK/72 10100 – MCLK/5 10101 – MCLK/10 10110 – MCLK/15 10111 – MCLK/17 11000 – MCLK/20 11001 – MCLK/25 11010 – MCLK/30 11011 – MCLK/32 11100 – MCLK/34 Others – MCLK/4

6.2 MCLK / LRCK RATIO AND MCLK / SCLK RATIO

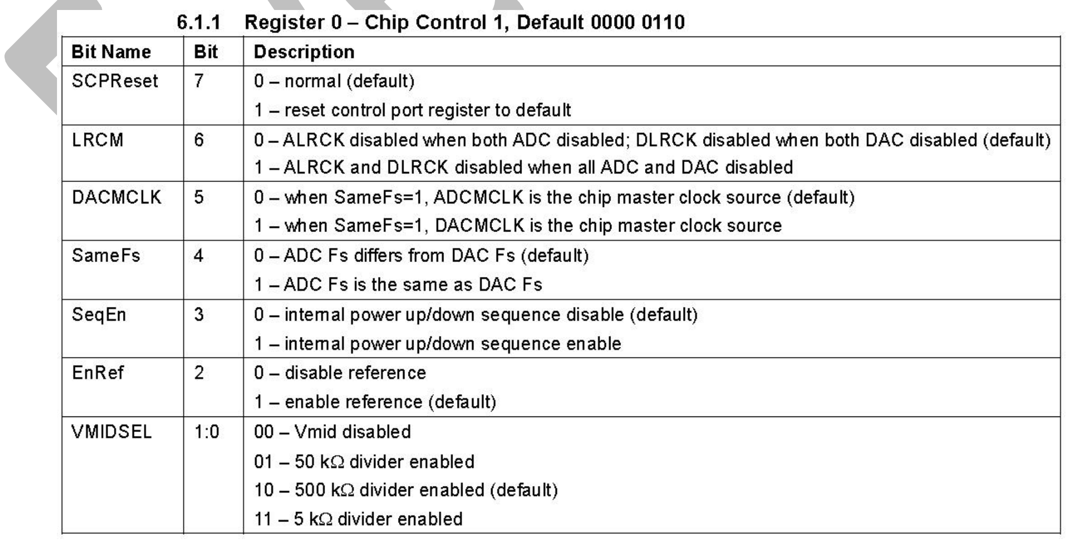
In master mode, the SCLK and LRCK signals are generated by ES8388 when any of the ADCs or DACs is enabled.

In slave mode, the SCLK and LRCK clock outputs are disabled by default to allow another digital audio interface to drive these pins.

ES8388 include stereo ADC and stereo DAC. The ADC and DAC have individual LRCK clock control bit even if there is only one LRCK pin in ES8388. **Thus the ALRCK and DLRCK must be same in ES8388 internally. Bit7(slrck) in register 43 should be used to select the same LRCK or individual LRCK for ADC and DAC.**

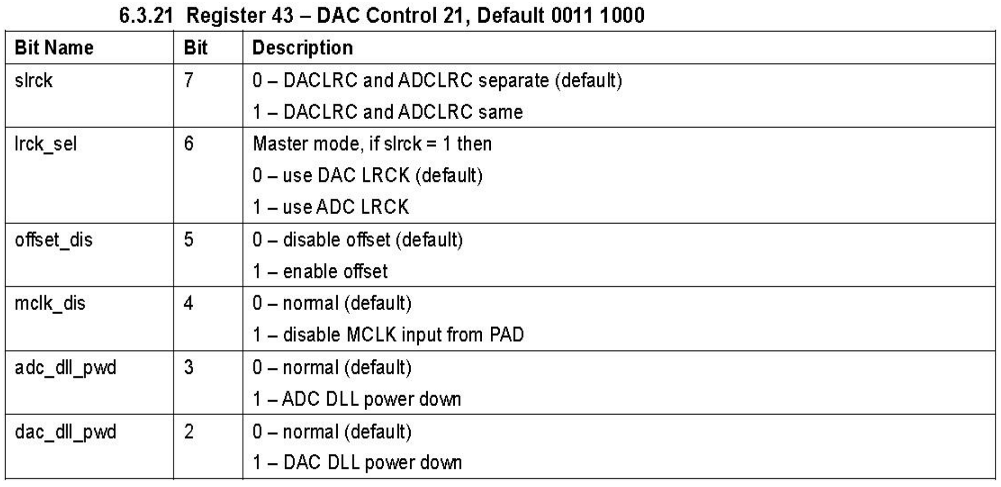
If ADC LRCK is used as internal LRCK, the Bit6 of register43 should be 1, Bit 5 of register0 should be 0 and Bit4 of registrer0 should be 1.

If DAC LRCK is used as internal LRCK, the Bit6 of register43 should be 0, Bit 5 of register0 should be 1 and Bit4 of registrer0 should be 1.



6.1.1 Register 0 – Chip Control 1, Default 0000 0110

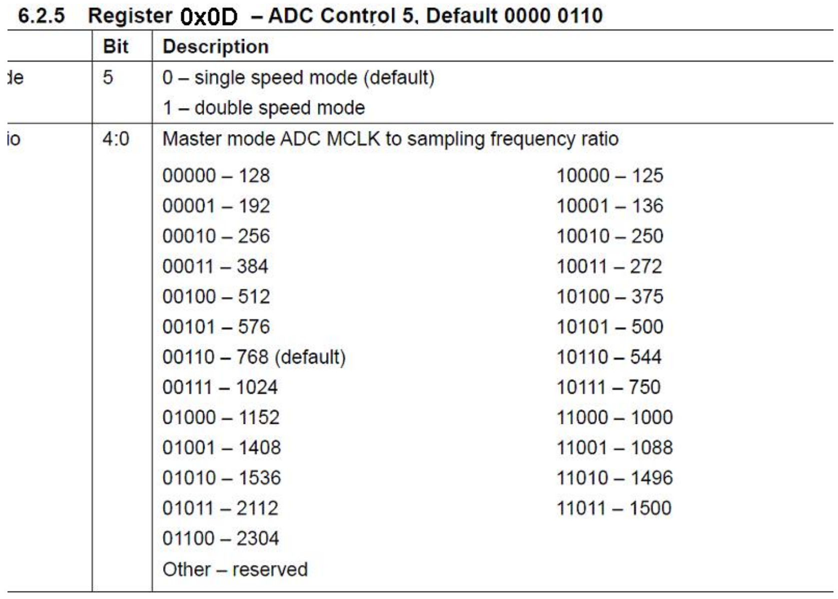
Bit Name	Bit	Description
SCPReset	7	0 – normal (default) 1 – reset control port register to default
LRCM	6	0 – ALRCK disabled when both ADC disabled; DLRCK disabled when both DAC disabled (default) 1 – ALRCK and DLRCK disabled when all ADC and DAC disabled
DACMCLK	5	0 – when SameFs=1, ADCMCLK is the chip master clock source (default) 1 – when SameFs=1, DACMCLK is the chip master clock source
SameFs	4	0 – ADC Fs differs from DAC Fs (default) 1 – ADC Fs is the same as DAC Fs
SeqEn	3	0 – internal power up/down sequence disable (default) 1 – internal power up/down sequence enable
EnRef	2	0 – disable reference 1 – enable reference (default)
VMIDSEL	1:0	00 – Vmid disabled 01 – 50 kΩ divider enabled 10 – 500 kΩ divider enabled (default) 11 – 5 kΩ divider enabled



6.3.21 Register 43 – DAC Control 21, Default 0011 1000

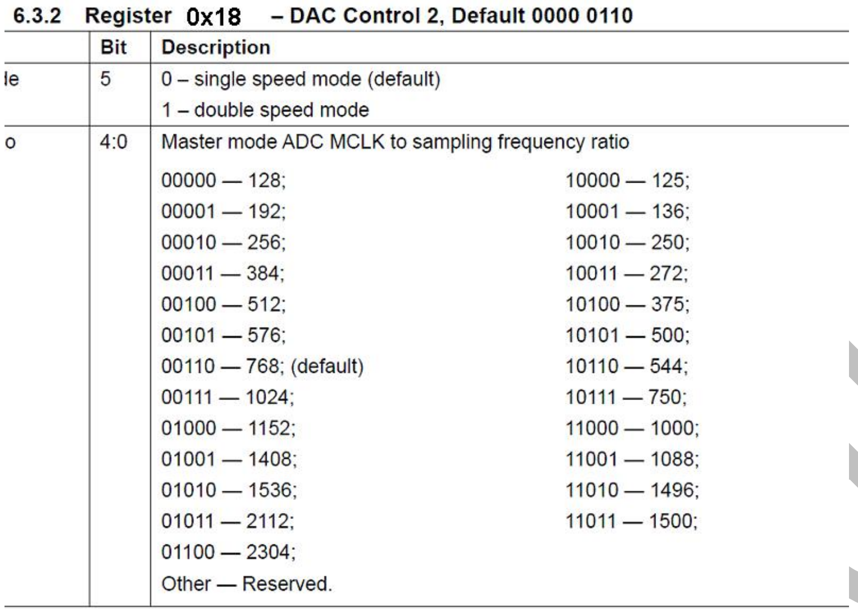
Bit Name	Bit	Description
slrck	7	0 – DACLRC and ADCLRC separate (default) 1 – DACLRC and ADCLRC same
lrck_sel	6	Master mode, if slrck = 1 then 0 – use DAC LRCK (default) 1 – use ADC LRCK
offset_dis	5	0 – disable offset (default) 1 – enable offset
mclk_dis	4	0 – normal (default) 1 – disable MCLK input from PAD
adc_dll_pwd	3	0 – normal (default) 1 – ADC DLL power down
dac_dll_pwd	2	0 – normal (default) 1 – DAC DLL power down

Register 0x0D is used for ADC MCLK/LRCK ratio setting. Register 0x18 is used for DAC MCLK/LRCK ratio setting. Please refer to the MCLK/LRCK table listed below.



6.2.5 Register 0x0D – ADC Control 5, Default 0000 0110

	Bit	Description	
de	5	0 – single speed mode (default) 1 – double speed mode	
io	4:0	Master mode ADC MCLK to sampling frequency ratio	
		00000 – 128	10000 – 125
		00001 – 192	10001 – 136
		00010 – 256	10010 – 250
		00011 – 384	10011 – 272
		00100 – 512	10100 – 375
		00101 – 576	10101 – 500
		00110 – 768 (default)	10110 – 544
		00111 – 1024	10111 – 750
		01000 – 1152	11000 – 1000
		01001 – 1408	11001 – 1088
		01010 – 1536	11010 – 1496
		01011 – 2112	11011 – 1500
		01100 – 2304	
		Other – reserved	



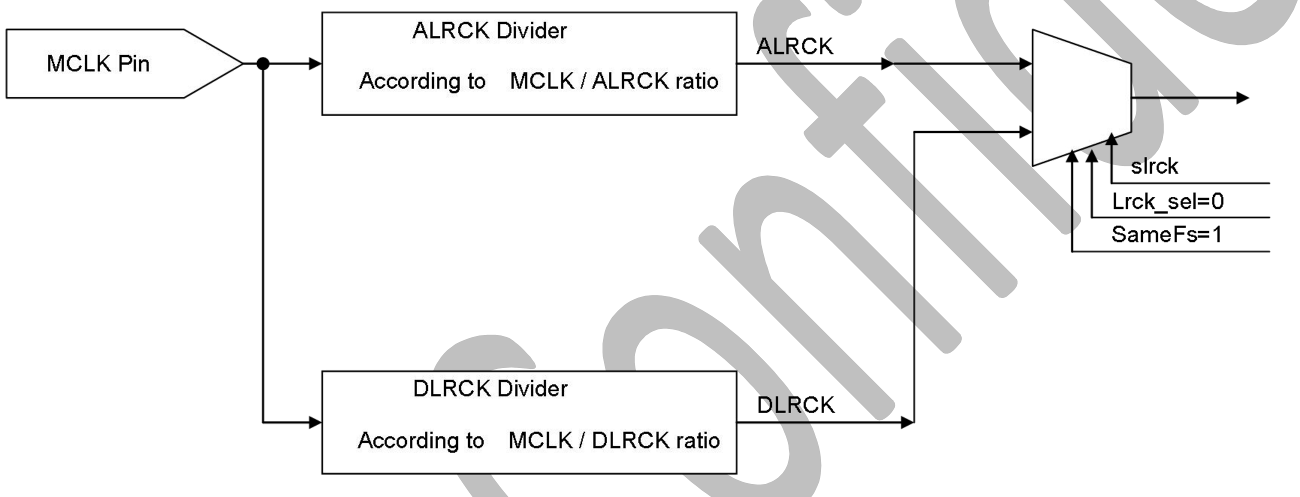
6.3.2 Register 0x18 – DAC Control 2, Default 0000 0110

	Bit	Description	
le	5	0 – single speed mode (default) 1 – double speed mode	
o	4:0	Master mode ADC MCLK to sampling frequency ratio	
		00000 — 128;	10000 — 125;
		00001 — 192;	10001 — 136;
		00010 — 256;	10010 — 250;
		00011 — 384;	10011 — 272;
		00100 — 512;	10100 — 375;
		00101 — 576;	10101 — 500;
		00110 — 768; (default)	10110 — 544;
		00111 — 1024;	10111 — 750;
		01000 — 1152;	11000 — 1000;
		01001 — 1408;	11001 — 1088;
		01010 — 1536;	11010 — 1496;
		01011 — 2112;	11011 — 1500;
		01100 — 2304;	
		Other — Reserved.	

In slave mode, ES8388 will auto-check MCLK/LRCK ratio. **Only the left side ratios in the table are supported in slave mode**.

In master mode, all ratios in the table are supported. Codec send the LRCK to external CPU/DSP/MCU according to MCLK / LRCK ratio setting. The value of register 0x0D and register 0x18 must be suitable for LRCK frequency.

Please refer to the diagram below to learn about the MCLK / LRCK ratio.



MCLK / LRCK ratio vs. LRCK Frequency in Master mode

The BCLKDIV control bits in register 8 are used for MCLK/SCLK ratio setting. Please refer to the BCLKDIV table listed below.

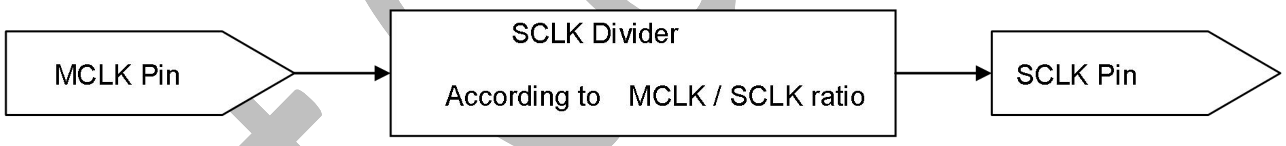
6.1.9 Register 8 – Master Mode Control, Default 1000 0000

Bit Name	Bit	Description
MSC	7	0 – slave serial port mode 1 – master serial port mode (default)
MCLKDIV2	6	0 – MCLK not divide (default) 1 – MCLK divide by 2
BCLK_INV	5	0 – normal (default) 1 – BCLK inverted
BCLKDIV	4:0	00000 – master mode BCLK generated automatically based on the clock table (default) 00001 – MCLK/1 00010 – MCLK/2 00011 – MCLK/3 00100 – MCLK/4 00101 – MCLK/6 00110 – MCLK/8 00111 – MCLK/9 01000 – MCLK/11 01001 – MCLK/12 01010 – MCLK/16 01011 – MCLK/18 01100 – MCLK/22 01101 – MCLK/24 01110 – MCLK/33 01111 – MCLK/36 10000 – MCLK/44 10001 – MCLK/48 10010 – MCLK/66 10011 – MCLK/72 10100 – MCLK/5 10101 – MCLK/10 10110 – MCLK/15 10111 – MCLK/17 11000 – MCLK/20 11001 – MCLK/25 11010 – MCLK/30 11011 – MCLK/32 11100 – MCLK/34 Others – MCLK/4

In slave mode, Codec will auto-check the MCLK/SCLK ratio in slave mode. Thus BCLKDIV control bits in register 0x08 should be "00000" in slave mode.

In master mode, Codec send SCLK signal to external CPU/DSP/MCU according to MCLK/SCLK ratio setting.

If the SFI of codec is I2S-24bit, the frequency of SCLK is usually equal to 64 or 48 times frequency of LRCK. If the SFI of codec is I2S-16bit, the frequency of SCLK is usually equal to 32 times frequency of LRCK.



MCLK / SCLK ratio vs. SCLK Frequency in Master mode

	Digital Audio Interface Format	SCLK Frequency (Minimum)	BCLKDIV(Maximun)
Master Mode	I2S / Left Justified / Right Justified / DSP -16bit	SCLK = 32 x LRCK	Integer of MCLK/SCLK
	I2S / Left Justified / Right Justified / DSP -18bit	SCLK = 36 x LRCK	
	I2S / Left Justified / Right Justified / DSP -20bit	SCLK = 40 x LRCK	
	I2S / Left Justified / Right Justified / DSP -24bit	SCLK = 48 x LRCK	
	I2S / Left Justified / Right Justified / DSP -32bit	SCLK = 64 x LRCK	

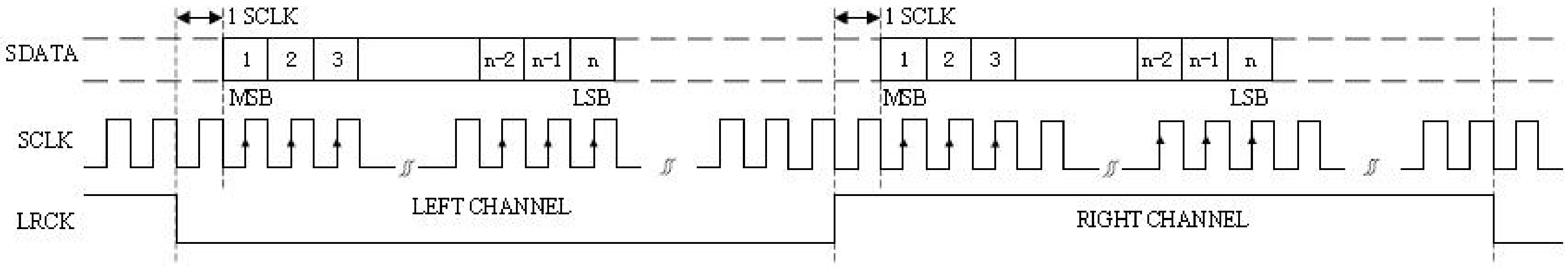
6.3 FOUR DIGITAL AUDIO FORMATS

ES8388 supports 4 digital audio formats.

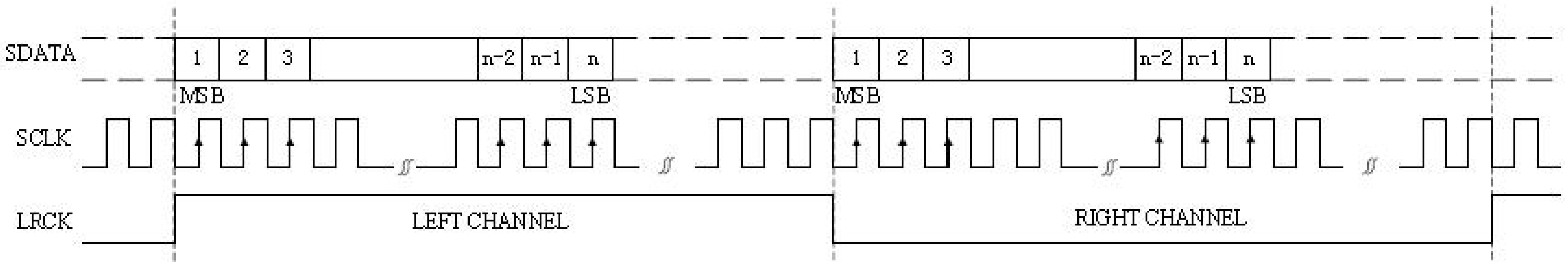
- ✧ **I2S**
- ✧ **Left Justified**
- ✧ **Right Justified**
- ✧ **DSP Mode (PCM)**

All of these four formats are MSB first.

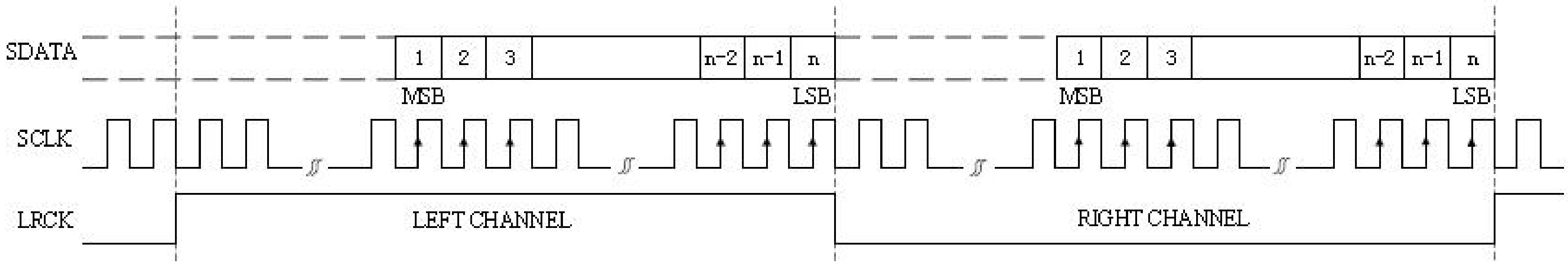
The register 12 and register 23 are used for ADC and DAC formats.



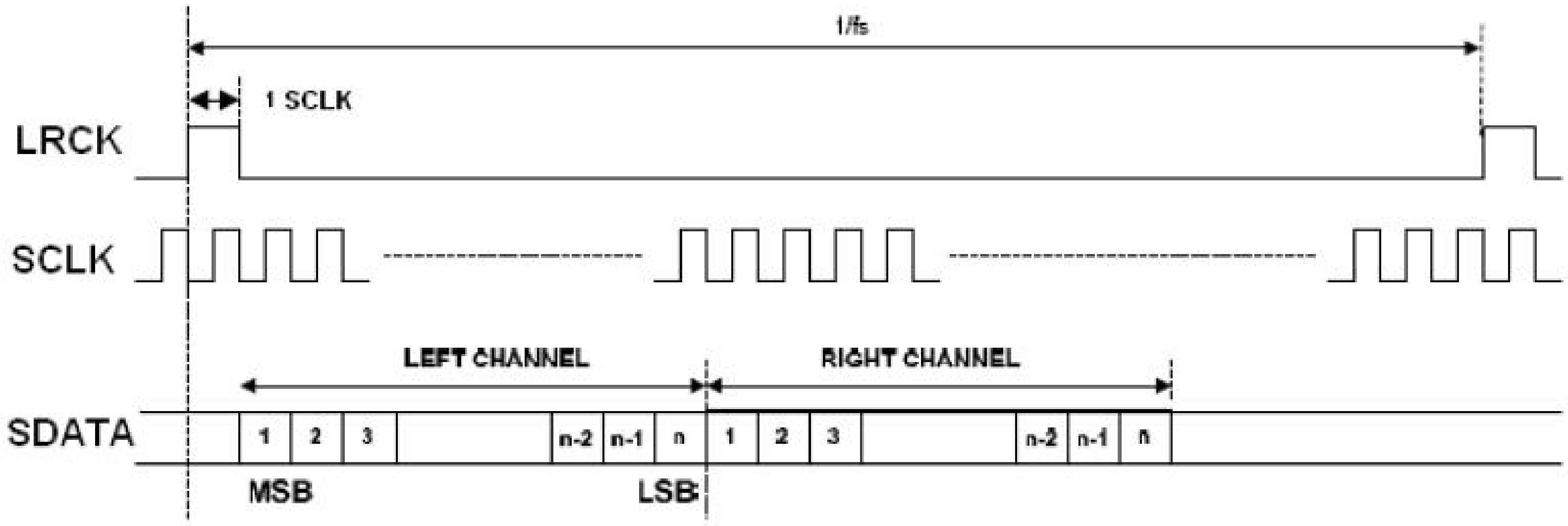
I^{2}S Serial Audio Data Format Up To 24-bit



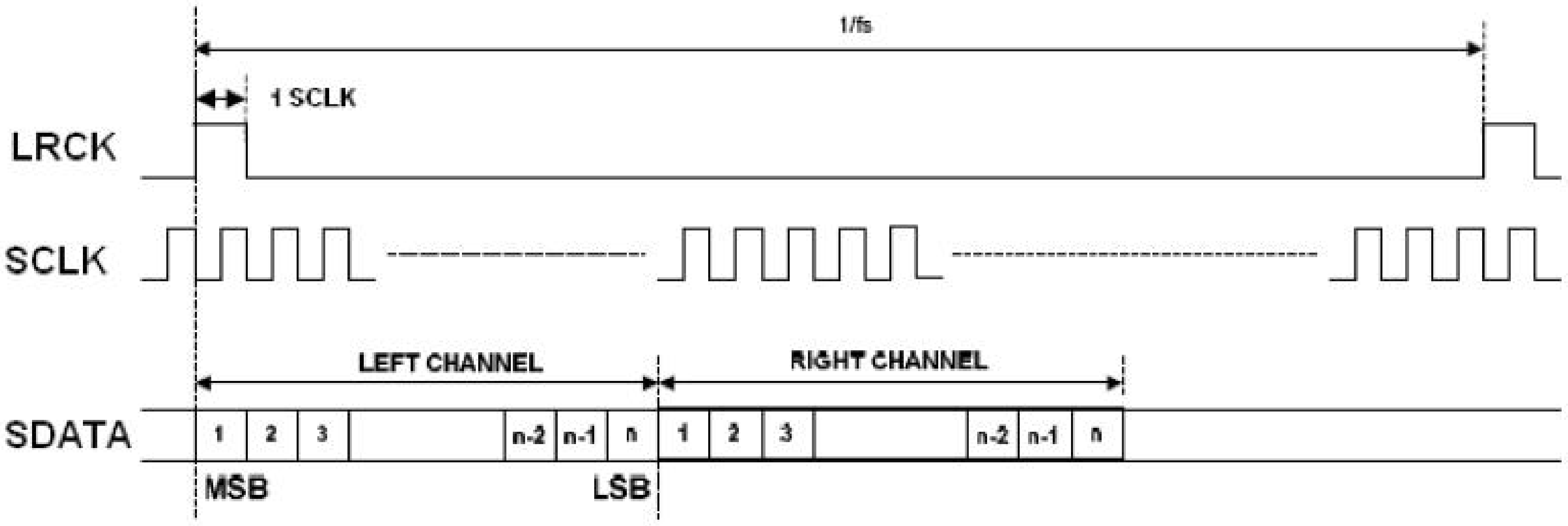
Left Justified Serial Audio Data Format Up To 24-bit



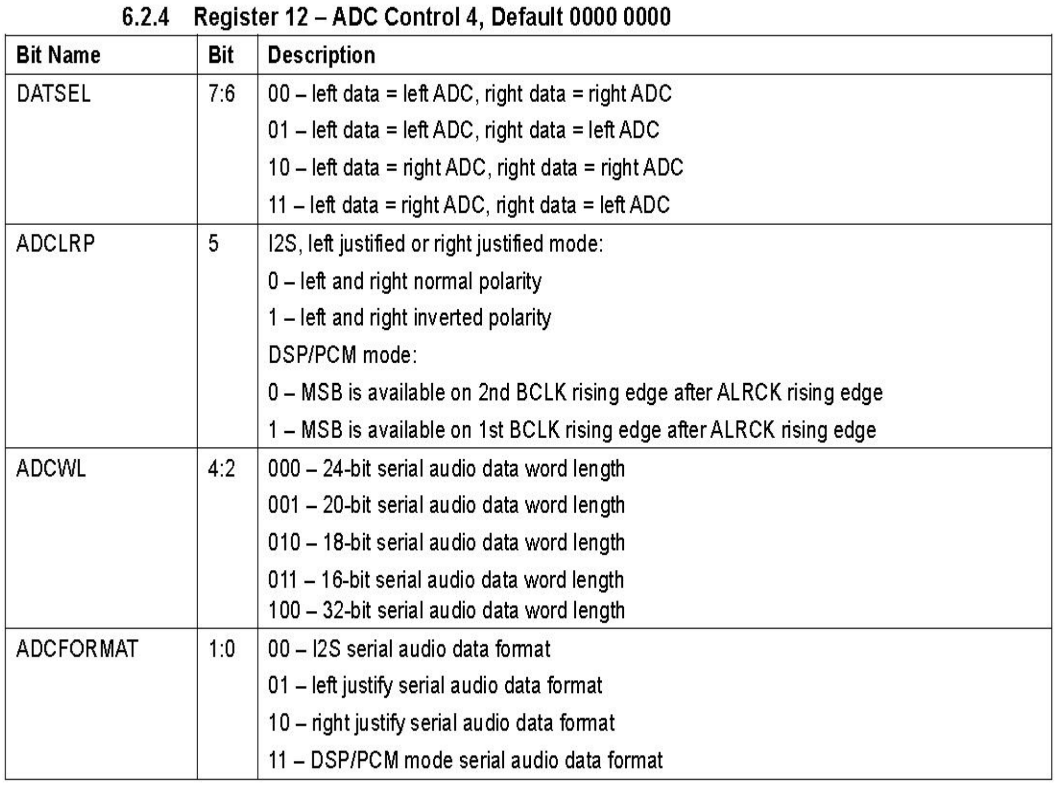
Right Justified Serial Audio Data Format Up To 24-bit



DSP (PCM) Mode A

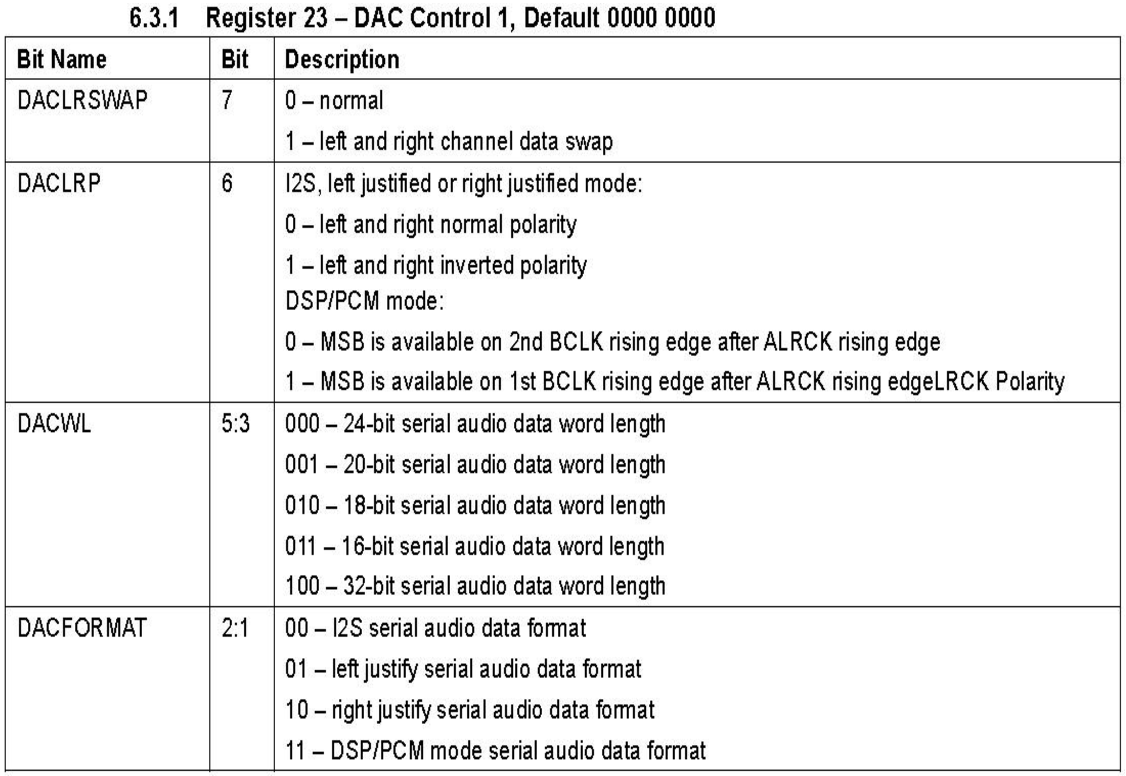


DSP (PCM) Mode B



6.2.4 Register 12 – ADC Control 4, Default 0000 0000

Bit Name	Bit	Description
DATSEL	7:6	00 – left data = left ADC, right data = right ADC 01 – left data = left ADC, right data = left ADC 10 – left data = right ADC, right data = right ADC 11 – left data = right ADC, right data = left ADC
ADCLRP	5	I2S, left justified or right justified mode: 0 – left and right normal polarity 1 – left and right inverted polarity DSP/PCM mode: 0 – MSB is available on 2nd BCLK rising edge after ALRCK rising edge 1 – MSB is available on 1st BCLK rising edge after ALRCK rising edge
ADCWL	4:2	000 – 24-bit serial audio data word length 001 – 20-bit serial audio data word length 010 – 18-bit serial audio data word length 011 – 16-bit serial audio data word length 100 – 32-bit serial audio data word length
ADCFORMAT	1:0	00 – I2S serial audio data format 01 – left justify serial audio data format 10 – right justify serial audio data format 11 – DSP/PCM mode serial audio data format



6.3.1 Register 23 – DAC Control 1, Default 0000 0000

Bit Name	Bit	Description
DACLRSWAP	7	0 – normal 1 – left and right channel data swap
DACLRP	6	I2S, left justified or right justified mode: 0 – left and right normal polarity 1 – left and right inverted polarity DSP/PCM mode: 0 – MSB is available on 2nd BCLK rising edge after ALRCK rising edge 1 – MSB is available on 1st BCLK rising edge after ALRCK rising edgeLRCK Polarity
DACWL	5:3	000 – 24-bit serial audio data word length 001 – 20-bit serial audio data word length 010 – 18-bit serial audio data word length 011 – 16-bit serial audio data word length 100 – 32-bit serial audio data word length
DACFORMAT	2:1	00 – I2S serial audio data format 01 – left justify serial audio data format 10 – right justify serial audio data format 11 – DSP/PCM mode serial audio data format

7 CHIP CONTROL AND POWER MANAGEMENT REGISTER

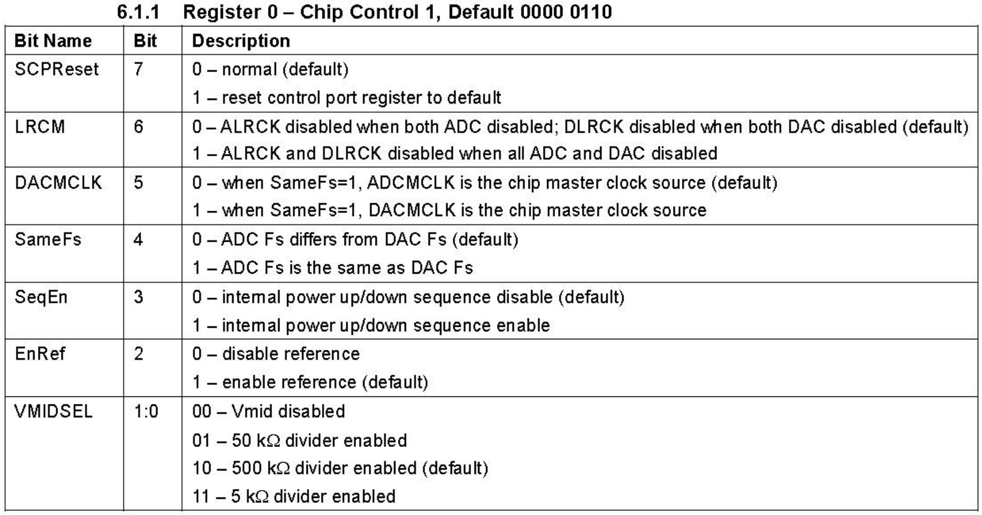
Some registers are used to ES8388 power management. These register's address located from register 0 to register 6.

The register 0 and register 1 should be used to control the internal bias current and internal reference voltage of ES8388. The control bits in these two registers must be enabled when ES8388 start up.

The register 2 should be used to control the digital block, state machine, DLL and internal reference voltage. The bits in register 2 must be zero when ES8388 start up.

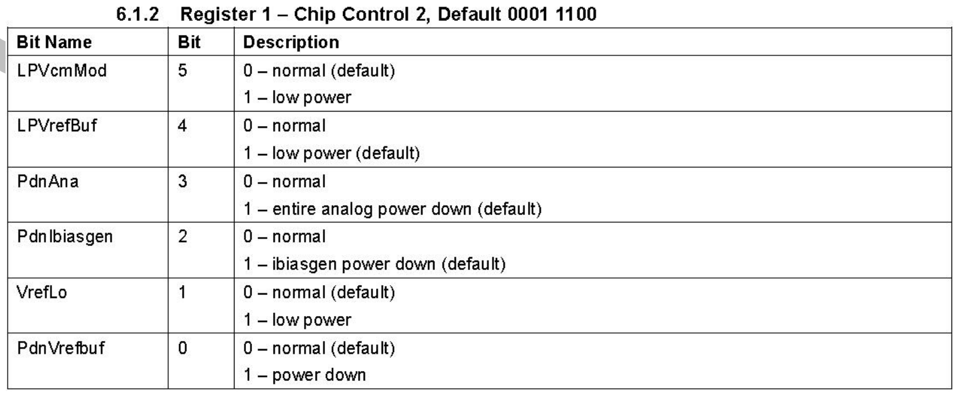
The register 3 should be used to power up ADC and Line inputs. The register 3 should be used to power up DAC and Line outputs.

The register 5 and register 6 should be used to set low power mode.



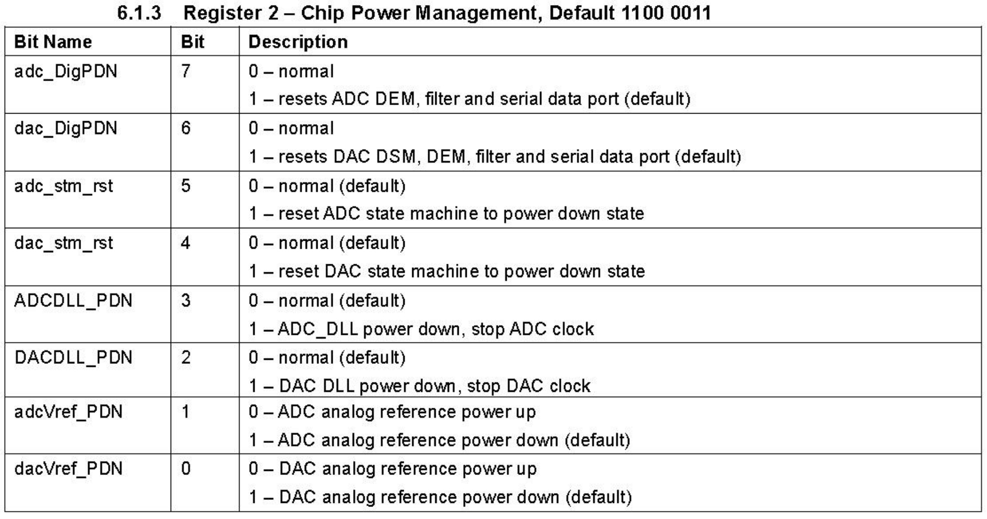
6.1.1 Register 0 – Chip Control 1, Default 0000 0110

Bit Name	Bit	Description
SCPReset	7	0 – normal (default) 1 – reset control port register to default
LRCM	6	0 – ALRCK disabled when both ADC disabled; DLRCK disabled when both DAC disabled (default) 1 – ALRCK and DLRCK disabled when all ADC and DAC disabled
DACMCLK	5	0 – when SameFs=1, ADCMCLK is the chip master clock source (default) 1 – when SameFs=1, DACMCLK is the chip master clock source
SameFs	4	0 – ADC Fs differs from DAC Fs (default) 1 – ADC Fs is the same as DAC Fs
SeqEn	3	0 – internal power up/down sequence disable (default) 1 – internal power up/down sequence enable
EnRef	2	0 – disable reference 1 – enable reference (default)
VMIDSEL	1:0	00 – Vmid disabled 01 – 50 kΩ divider enabled 10 – 500 kΩ divider enabled (default) 11 – 5 kΩ divider enabled



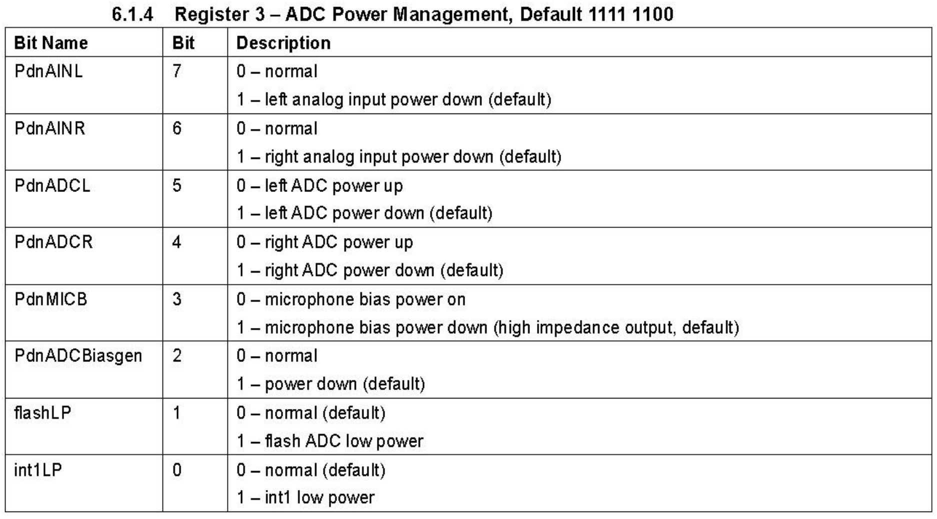
6.1.2 Register 1 – Chip Control 2, Default 0001 1100

Bit Name	Bit	Description
LPVcmMod	5	0 – normal (default) 1 – low power
LPVrefBuf	4	0 – normal 1 – low power (default)
PdnAna	3	0 – normal 1 – entire analog power down (default)
PdnIbiasgen	2	0 – normal 1 – ibiasgen power down (default)
VrefLo	1	0 – normal (default) 1 – low power
PdnVrefbuf	0	0 – normal (default) 1 – power down



6.1.3 Register 2 – Chip Power Management, Default 1100 0011

Bit Name	Bit	Description
adc_DigPDN	7	0 – normal 1 – resets ADC DEM, filter and serial data port (default)
dac_DigPDN	6	0 – normal 1 – resets DAC DSM, DEM, filter and serial data port (default)
adc_stm_rst	5	0 – normal (default) 1 – reset ADC state machine to power down state
dac_stm_rst	4	0 – normal (default) 1 – reset DAC state machine to power down state
ADCDLL_PDN	3	0 – normal (default) 1 – ADC_DLL power down, stop ADC clock
DACDLL_PDN	2	0 – normal (default) 1 – DAC DLL power down, stop DAC clock
adcVref_PDN	1	0 – ADC analog reference power up 1 – ADC analog reference power down (default)
dacVref_PDN	0	0 – DAC analog reference power up 1 – DAC analog reference power down (default)



6.1.4 Register 3 – ADC Power Management, Default 1111 1100

Bit Name	Bit	Description
PdnAINL	7	0 – normal 1 – left analog input power down (default)
PdnAINR	6	0 – normal 1 – right analog input power down (default)
PdnADCL	5	0 – left ADC power up 1 – left ADC power down (default)
PdnADCR	4	0 – right ADC power up 1 – right ADC power down (default)
PdnMICB	3	0 – microphone bias power on 1 – microphone bias power down (high impedance output, default)
PdnADCBiasgen	2	0 – normal 1 – power down (default)
flashLP	1	0 – normal (default) 1 – flash ADC low power
int1LP	0	0 – normal (default) 1 – int1 low power

6.1.5 Register 4 – DAC Power Management, Default 1100 0000

Bit Name	Bit	Description
PdnDACL	7	0 – left DAC power up 1 – left DAC power down (default)
PdnDACR	6	0 – right DAC power up 1 – right DAC power down (default)
LOUT1	5	0 – LOUT1 disabled (default) 1 – LOUT1 enabled
ROUT1	4	0 – ROUT1 disabled (default) 1 – ROUT1 enabled
LOUT2	3	0 – LOUT2 disabled (default) 1 – LOUT2 enabled
ROUT2	2	0 – ROUT2 disabled (default) 1 – ROUT2 enabled

6.1.6 Register 5 – Chip Low Power 1, Default 0000 0000

Bit Name	Bit	Description
LPDACL	7	0 – normal (default) 1 – low power
LPDACR	6	0 – normal (default) 1 – low power
LPLOUT1	5	0 – normal (default) 1 – low power
LPLOUT2	3	0 – normal (default) 1 – low power

6.1.7 Register 6 – Chip Low Power 2, Default 0000 0000

Bit Name	Bit	Description
LPPGA	7	0 – normal (default) 1 – low power
LPLMIX	6	0 – normal (default) 1 – low power
LPADCvrp	1	0 – normal (default) 1 – low power
LPDACvrp	0	0 – normal (default) 1 – low power

8 ANALOG INPUT SIGNAL PATH

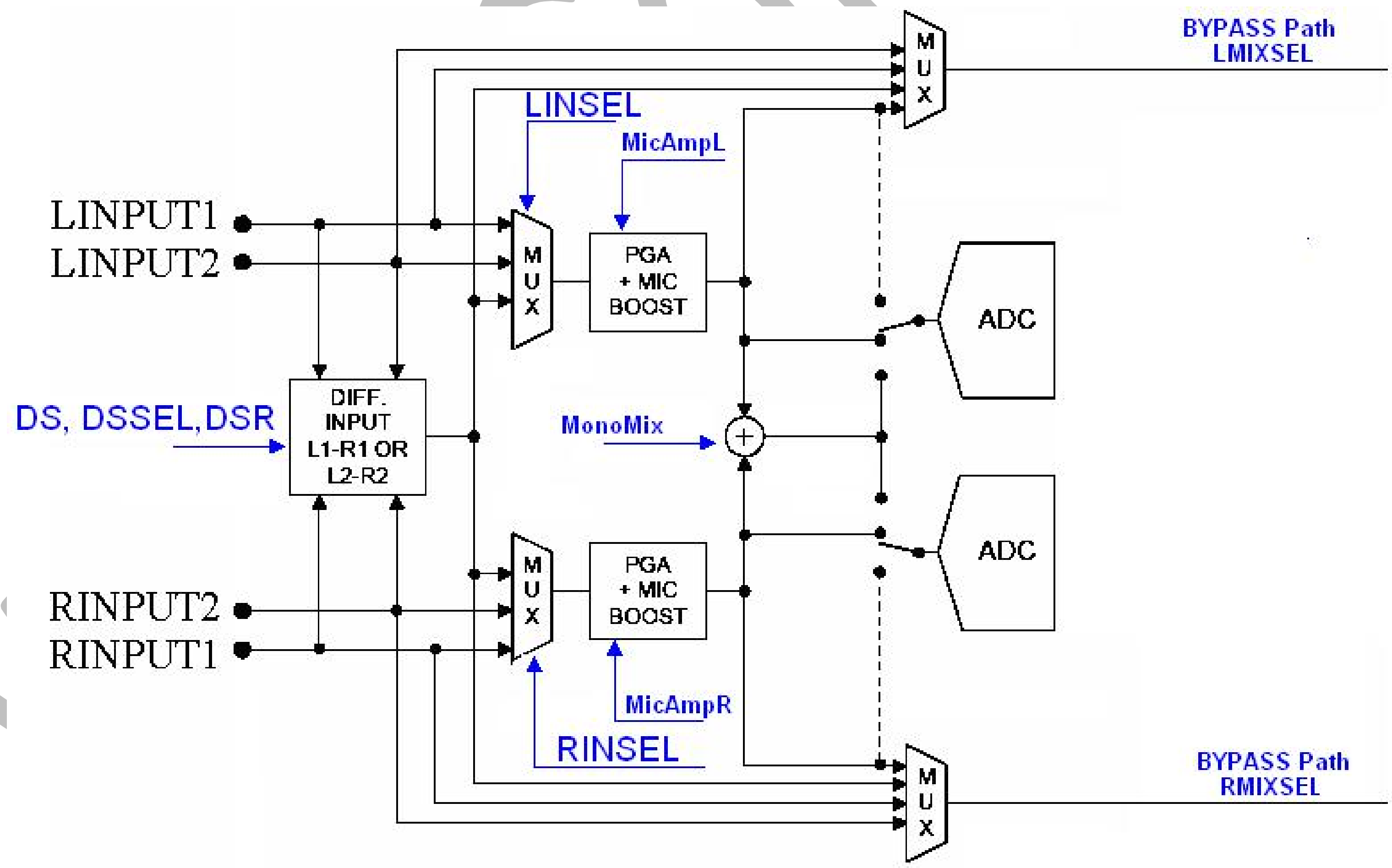
ES8388 has 4 analog input pins which may be used to support connections to multiple microphone or line input sources. The input multiplexers on the left and right channels can be used to select different configurations for each of the input sources. The analog input paths can support line inputs or microphone inputs, in single-ended mode, pseudo-differential and fully-differential modes. The input stage can also provide common mode noise rejection in some configurations.

The Left and Right analog input channels are routed to ADCs where it is digitised. Alternatively, the two channels can also be mixed in the analog domain and digitized in one ADC while the other ADC is switch off. The mono-mix signal appears on both digital output channels.

There is also a BYPASS path for each channel, enabling the analog input signal to be routed directly to the output multiplexers and PGAs.

8.1 THE ANALOG INPUT SIGNAL PATHS AND THE CONTROL REGISTER

The ES8388 analog input signal paths and control registers are illustrated in the figure which be listed below.



pdnAINL and pdnAINR control bits in register 3 should be used to power up / power down the input channel.

In this figure, LINSEL and RINSEL control bits are used to select independently between external inputs and internally generated differential products (LIN1-RIN1 or LIN2-RIN2). The choice of differential signal, LIN1-RIN1 or LIN2-RIN2 is made using DS, DSSEL and DSR control bits.

Example 1, the ES8388 can be set up to convert one differential signal by applying the differential signal to LIN1/RIN1. By

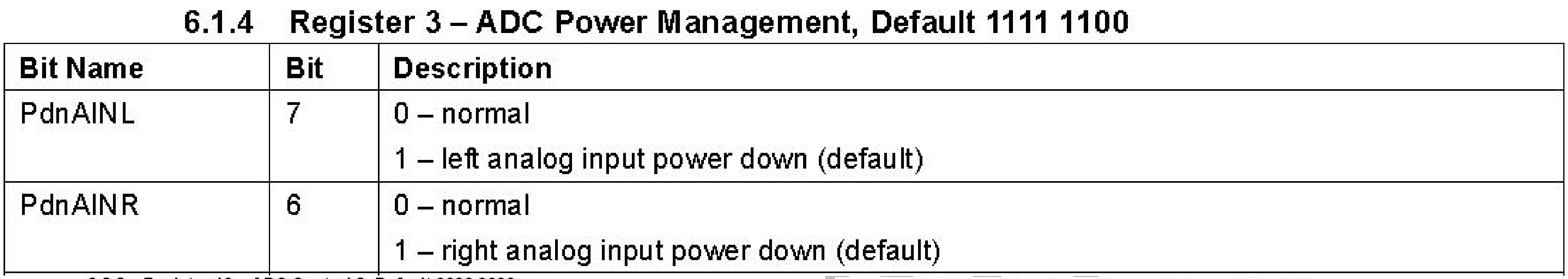
setting the LINSEL and RINSEL to L-R differential and setting the DSSEL, DSR and DS to zero, the differential signal can then be routed to the Left/Right ADCs.

Example 2, the ES8388 can be set up to convert two differential signal by applying one differential signal to LIN1/RIN1 and applying the other to LIN2/RIN2. By setting the LINSEL and RINSEL to L-R differential, setting the DSSEL and DSR to one and setting DS to zero, the LIN1/RIN1 differential signal can be routed to the Left ADC and the LIN2/RIN2 differential signal can be routed to Right ADC.

MicAmpL and MicAmpR control bits are used to control PGA to amplify the amplitude of input signals. The input signals are selected by LINSEL and RINSEL control bits. The PGA gain is adjustable from 0dB to +24dB in 3dB steps.

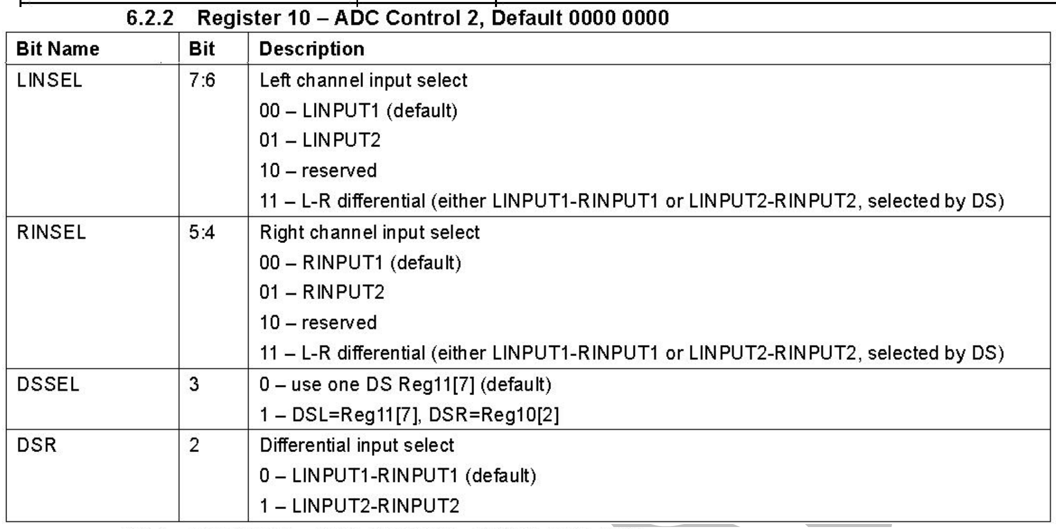
The two input channels can also be mixed in the analog domain and digitized in one ADC while the other ADC is switch off. By setting MONOMIX to ‘01’ or ‘10’, the mono-mix signal can then be routed to Left ADC or Right ADC. For analog mono mix either the left or right ADC can be used, allowing the unused ADC to be powered off. The user also has the flexibility to select the data output from the audio interface using DATSEL control bits.

LMIXSEL and RMIXSEL control bits are used to select the input channel of Left and Right Bypass path. The selected input channels can be routed directly to the the output multiplexers and PGAs.



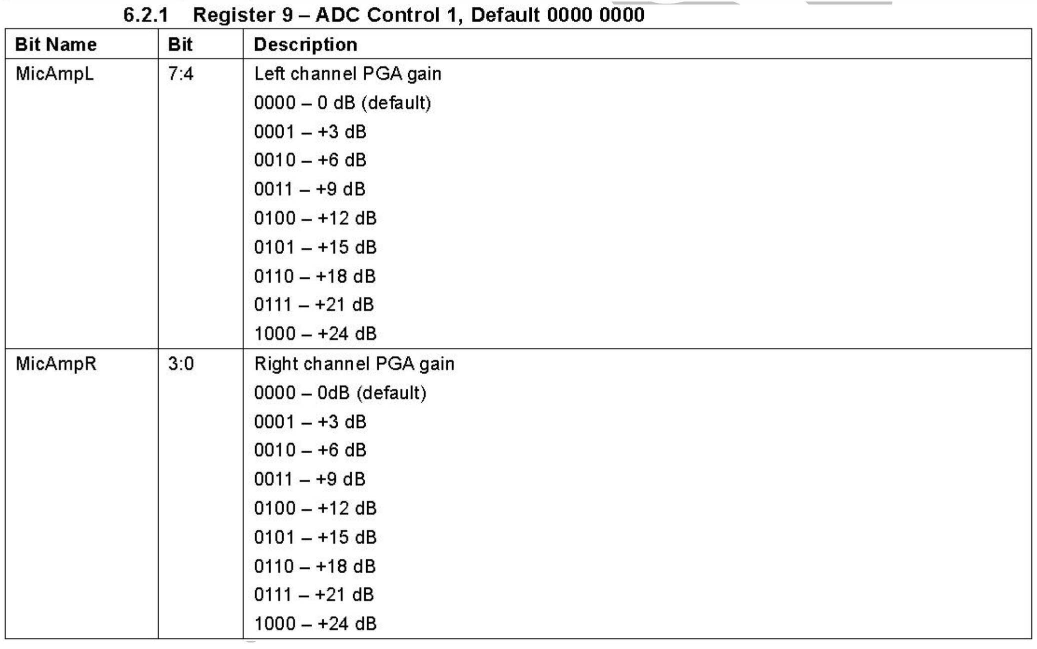
6.1.4 Register 3 – ADC Power Management, Default 1111 1100

Bit Name	Bit	Description
PdnAINL	7	0 – normal 1 – left analog input power down (default)
PdnAINR	6	0 – normal 1 – right analog input power down (default)



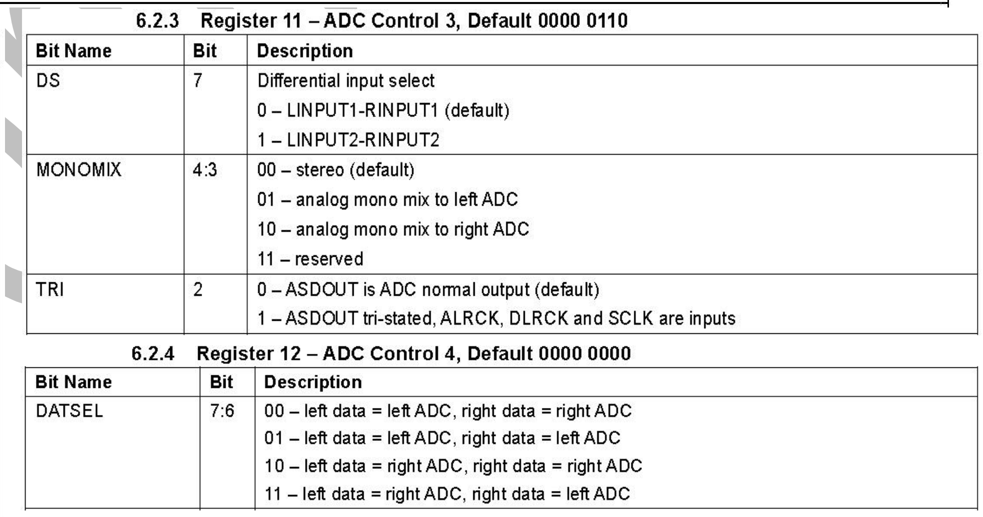
6.2.2 Register 10 – ADC Control 2, Default 0000 0000

Bit Name	Bit	Description
LINSEL	7:6	Left channel input select 00 – LINPUT1 (default) 01 – LINPUT2 10 – reserved 11 – L-R differential (either LINPUT1-RINPUT1 or LINPUT2-RINPUT2, selected by DS)
RINSEL	5:4	Right channel input select 00 – RINPUT1 (default) 01 – RINPUT2 10 – reserved 11 – L-R differential (either LINPUT1-RINPUT1 or LINPUT2-RINPUT2, selected by DS)
DSSEL	3	0 – use one DS Reg11[7] (default) 1 – DSL=Reg11[7], DSR=Reg10[2]
DSR	2	Differential input select 0 – LINPUT1-RINPUT1 (default) 1 – LINPUT2-RINPUT2



6.2.1 Register 9 – ADC Control 1, Default 0000 0000

Bit Name	Bit	Description
MicAmpL	7:4	Left channel PGA gain 0000 – 0 dB (default) 0001 – +3 dB 0010 – +6 dB 0011 – +9 dB 0100 – +12 dB 0101 – +15 dB 0110 – +18 dB 0111 – +21 dB 1000 – +24 dB
MicAmpR	3:0	Right channel PGA gain 0000 – 0dB (default) 0001 – +3 dB 0010 – +6 dB 0011 – +9 dB 0100 – +12 dB 0101 – +15 dB 0110 – +18 dB 0111 – +21 dB 1000 – +24 dB

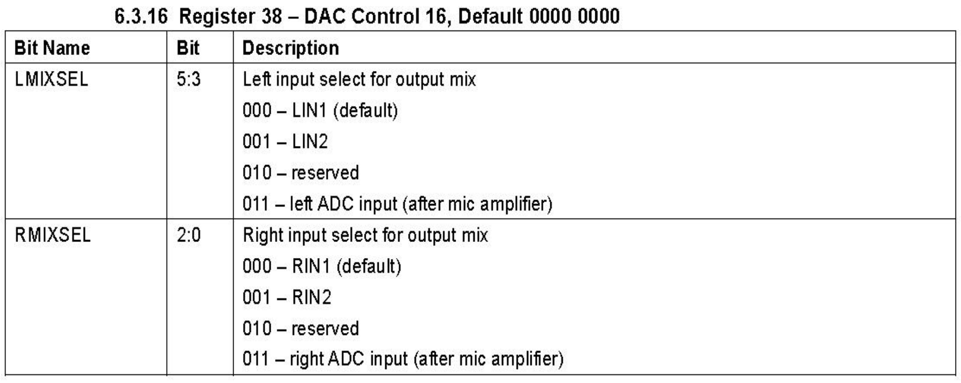


6.2.3 Register 11 – ADC Control 3, Default 0000 0110

Bit Name	Bit	Description
DS	7	Differential input select 0 – LINPUT1-RINPUT1 (default) 1 – LINPUT2-RINPUT2
MONOMIX	4:3	00 – stereo (default) 01 – analog mono mix to left ADC 10 – analog mono mix to right ADC 11 – reserved
TRI	2	0 – ASDOUT is ADC normal output (default) 1 – ASDOUT tri-stated, ALRCK, DLRCK and SCLK are inputs

6.2.4 Register 12 – ADC Control 4, Default 0000 0000

Bit Name	Bit	Description
DATSEL	7:6	00 – left data = left ADC, right data = right ADC 01 – left data = left ADC, right data = left ADC 10 – left data = right ADC, right data = right ADC 11 – left data = right ADC, right data = left ADC



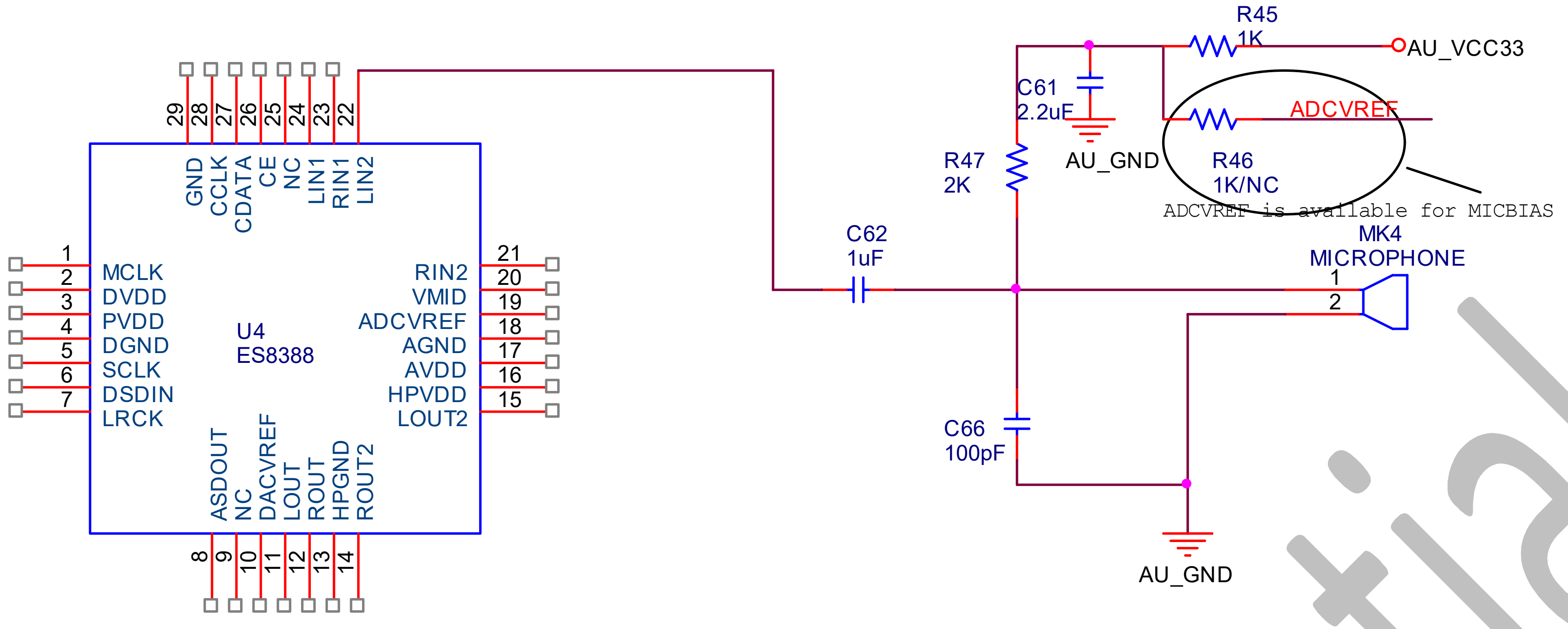
6.3.16 Register 38 – DAC Control 16, Default 0000 0000

Bit Name	Bit	Description
LMIXSEL	5:3	Left input select for output mix 000 – LIN1 (default) 001 – LIN2 010 – reserved 011 – left ADC input (after mic amplifier)
RMIXSEL	2:0	Right input select for output mix 000 – RIN1 (default) 001 – RIN2 010 – reserved 011 – right ADC input (after mic amplifier)

8.2 SINGLE-ENDED MICROPHONE INPUT

ES8388 should be used in single-ended microphone input mode when LINSEL and RINSEL in Register 10 don’t be set to ‘11’. The single-ended microphone input signal can be connected to LIN1, RIN1, LIN2 or RIN2.The MicAmpL or MicAmpR can be used to boost the microphone signal level.

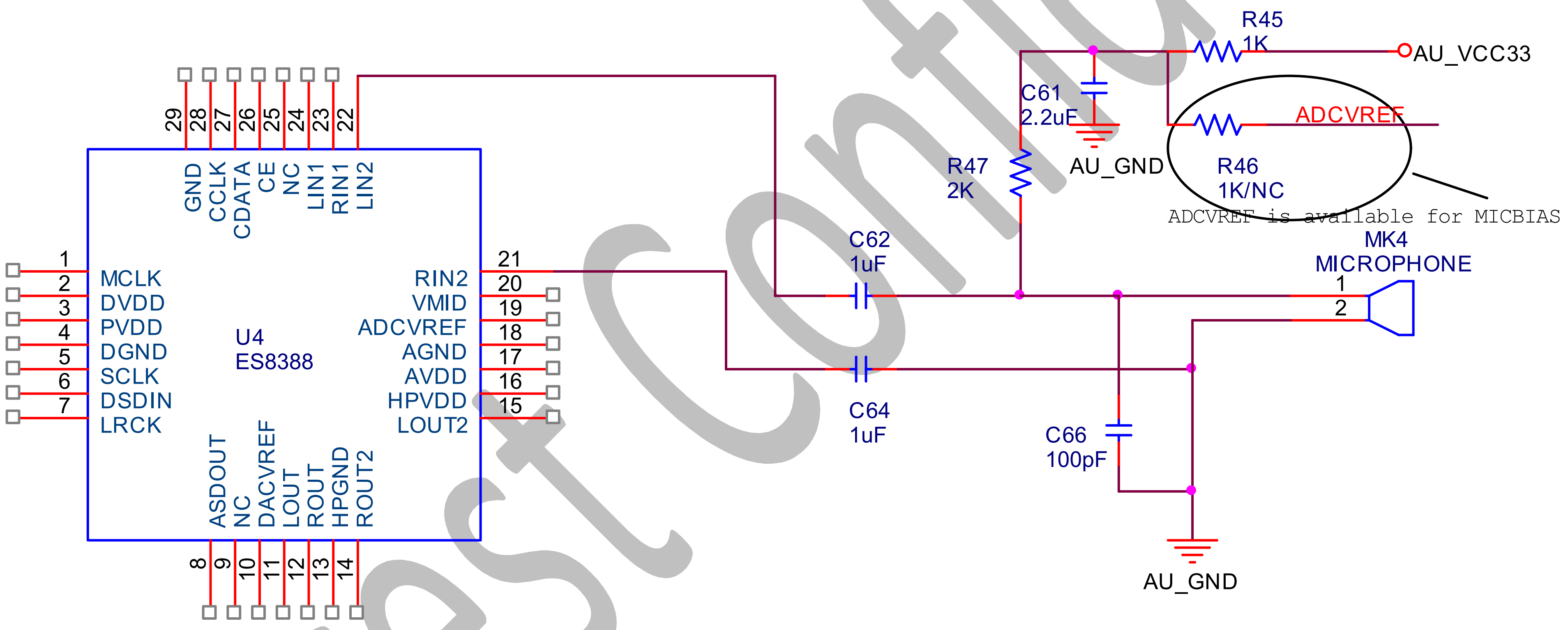
Single-ended microphone mode can not provide common mode noise rejection.



8.3 PESUDO-DIFFERENTIAL MICROPHONE INPUT

ES8388 should be used in pseudo-differential microphone input mode when LINSEL and RINSEL in Register 10 are set to ‘11’. The pseudo-differential microphone input signal can be connected to LIN1 and RIN1 or LIN2 and RIN2.The MicAmpL and MicAmpR can be used to boost the microphone signal level.

When GND has obvious noise, Pseudo-differential microphone mode can provide common mode noise rejection

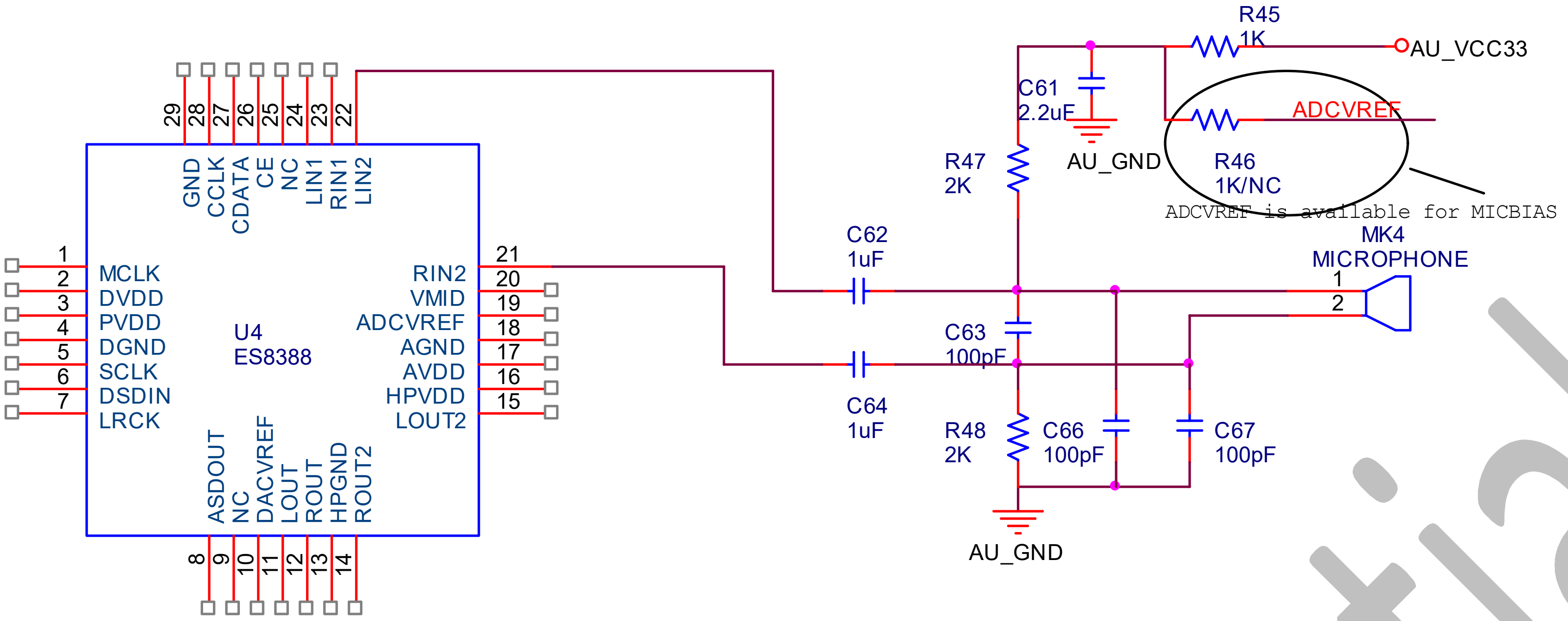


8.4 FULLY-DIFFERENTIAL MICROPHONE INPUT

ES8388 should be used in fully-differential microphone input mode when LINSEL and RINSEL in Register 10 are set to ‘11’b. The fully-differential microphone input signal can be connected to LIN1 and RIN1 or LIN2 and RIN2.The MicAmpL and MicAmpR can be used to boost the microphone signal level.

Fully-differential microphone mode can provide common mode noise rejection. The recording volume in fully-differential mode should be 2 times volume of pseudo-differential mode.

The fully-differential microphone mode is recommended for microphone recording.



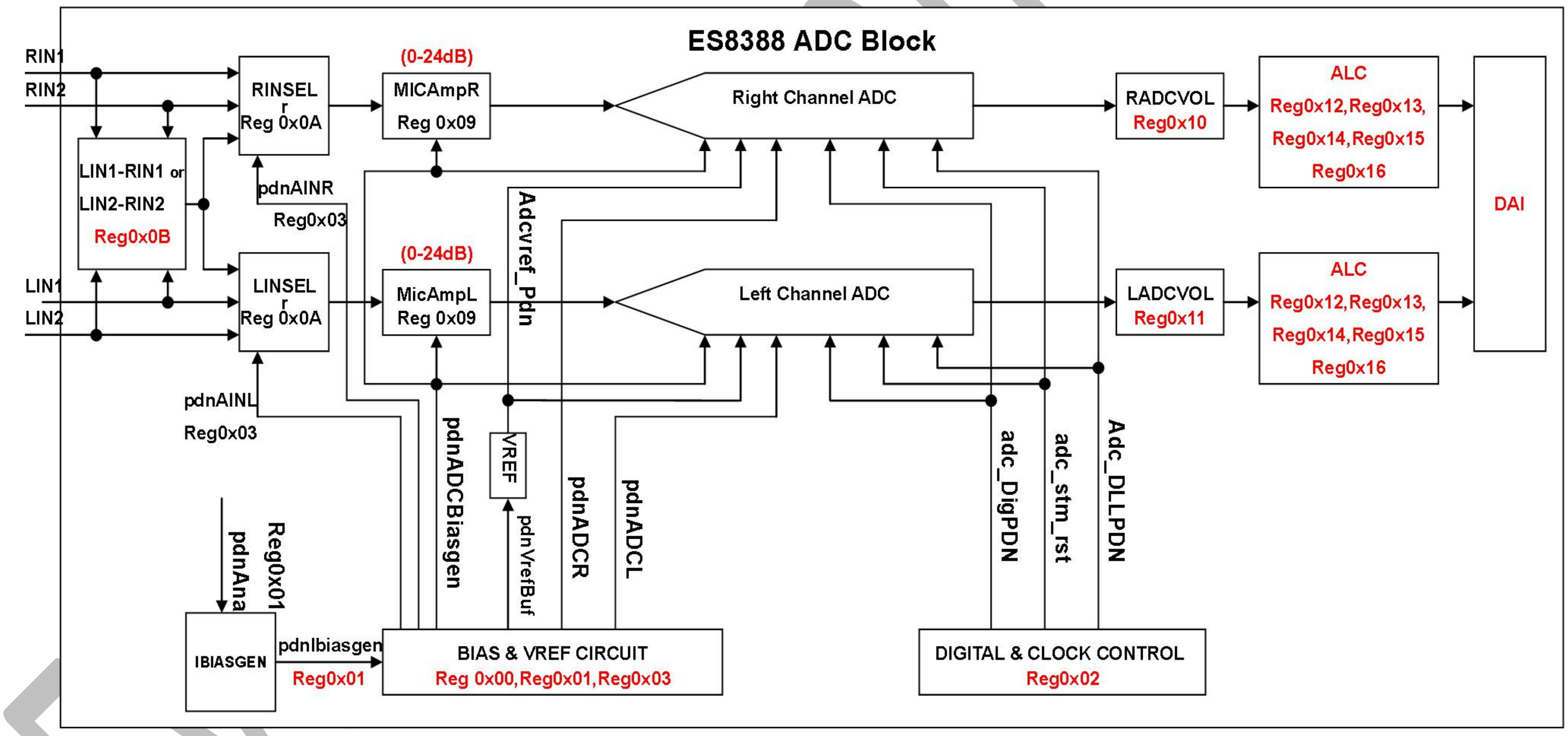
9 ADC FOR RECORDING

ES8388 can provides a stereo ADC for recording via I2S/PCM interface. The digital output data is sent out on ASDOUT pin.

The ADC full scale input level is proportional to AVDD. With a 3.3V supply voltage, the full scale level is 1.0Vrms. Any voltage greater than full scale may overload the ADC and cause distortion.

One ALC module should be used for recording. The registers used for ALC is located from register 18 to register 22.

9.1 THE ADC BLOCK DIAGRAM



9.2 THE ADC CONTROL REGISTERS

PdnADCL and PdnADCR control bits in register 3 should be used to power up / power down Left and right ADC.

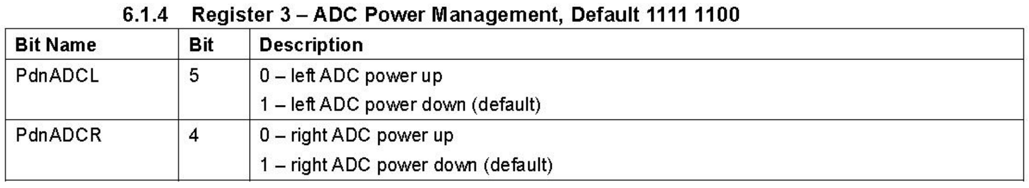
ADC_invL control bit in register 14 are used to invert the polarity of left channel ADC. ADC_invR control bit in register 14 are used for right channel ADC.

ADC_HPF_L and ADC_HPF_R control bits in register 14 are used to enable or disable high pass filter of left and right channel ADC. The default setting are recommended to ADC_HPF_L and ADC_HPF_R.

ADCSoftRamp control bit is used to fade in and fade out. ADCRampRate control bits are used to set the soft ramp rate.

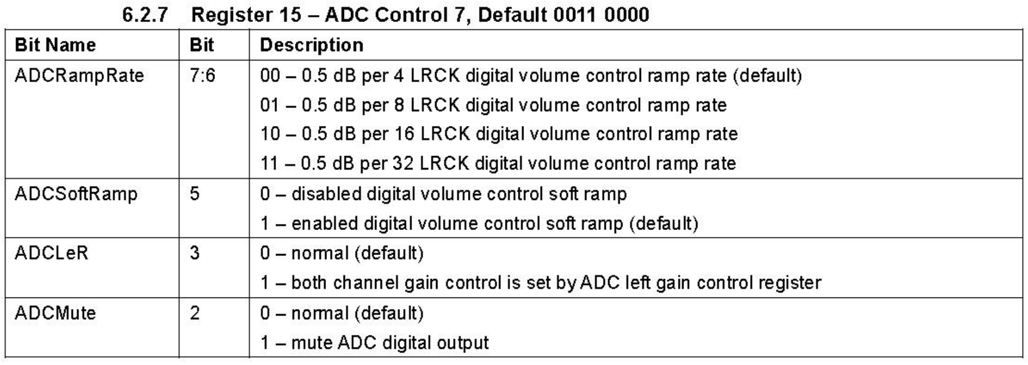
ES8388 ADC's digital volume can be adjustable from -96dB to 0dB in 0.5dB steps. LADCVOL and RADCVOL is used to independently control the left an right ADC recording volume. If ADCLeR control bit in register 15 is set to '1', the volume of left and right ADC can adjust synchronously by adjusting left ADC volume.

ES8388 ADC should be mute by setting ADCMute to '1'.



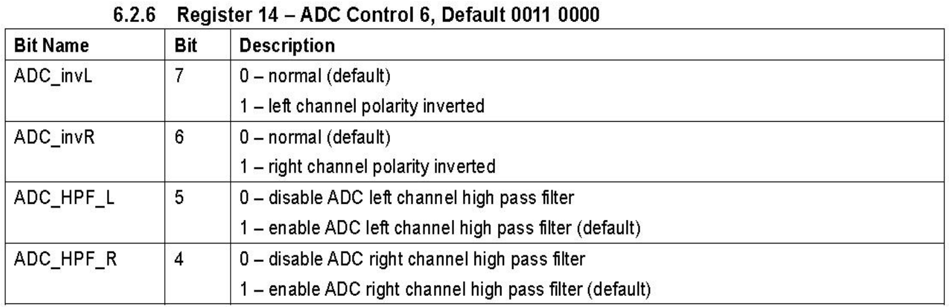
6.1.4 Register 3 – ADC Power Management, Default 1111 1100

Bit Name	Bit	Description
PdnADCL	5	0 – left ADC power up 1 – left ADC power down (default)
PdnADCR	4	0 – right ADC power up 1 – right ADC power down (default)



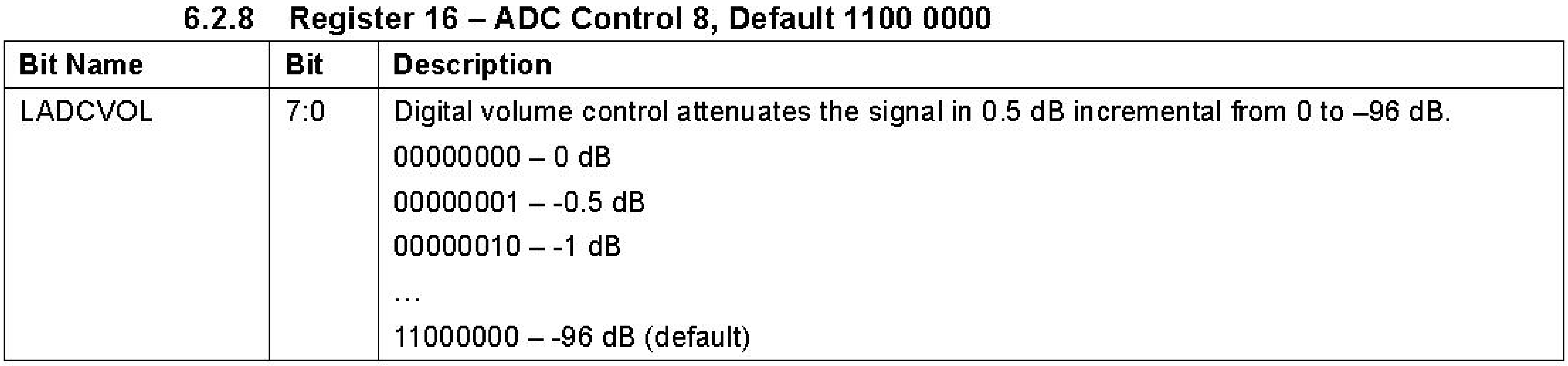
6.2.7 Register 15 – ADC Control 7, Default 0011 0000

Bit Name	Bit	Description
ADCRampRate	7:6	00 – 0.5 dB per 4 LRCK digital volume control ramp rate (default) 01 – 0.5 dB per 8 LRCK digital volume control ramp rate 10 – 0.5 dB per 16 LRCK digital volume control ramp rate 11 – 0.5 dB per 32 LRCK digital volume control ramp rate
ADCSoftRamp	5	0 – disabled digital volume control soft ramp 1 – enabled digital volume control soft ramp (default)
ADCLeR	3	0 – normal (default) 1 – both channel gain control is set by ADC left gain control register
ADCMute	2	0 – normal (default) 1 – mute ADC digital output



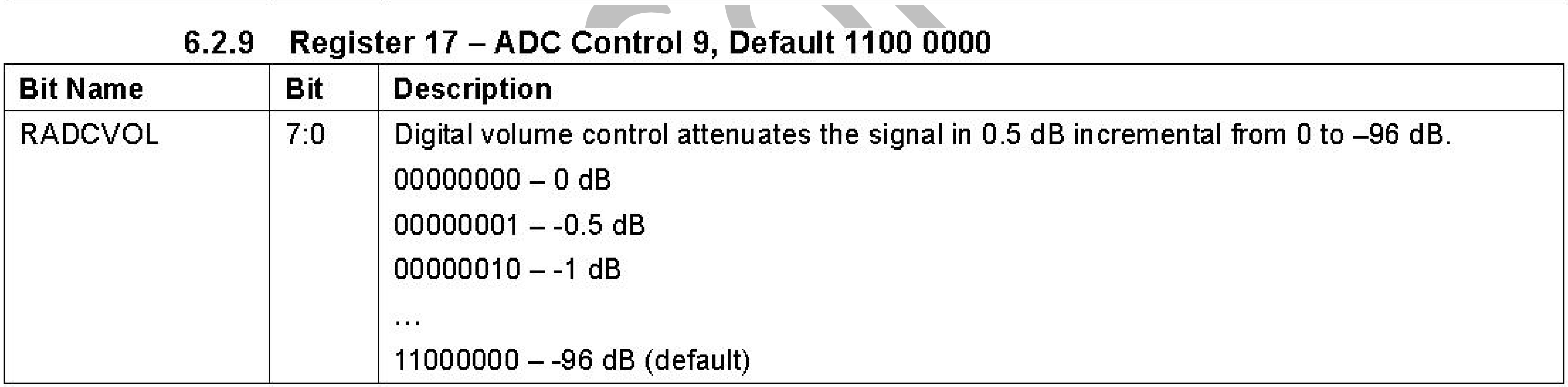
6.2.6 Register 14 – ADC Control 6, Default 0011 0000

Bit Name	Bit	Description
ADC_invL	7	0 – normal (default) 1 – left channel polarity inverted
ADC_invR	6	0 – normal (default) 1 – right channel polarity inverted
ADC_HPF_L	5	0 – disable ADC left channel high pass filter 1 – enable ADC left channel high pass filter (default)
ADC_HPF_R	4	0 – disable ADC right channel high pass filter 1 – enable ADC right channel high pass filter (default)



6.2.8 Register 16 – ADC Control 8, Default 1100 0000

Bit Name	Bit	Description
LADCVOL	7:0	Digital volume control attenuates the signal in 0.5 dB incremental from 0 to –96 dB. 00000000 – 0 dB 00000001 – -0.5 dB 00000010 – -1 dB … 11000000 – -96 dB (default)



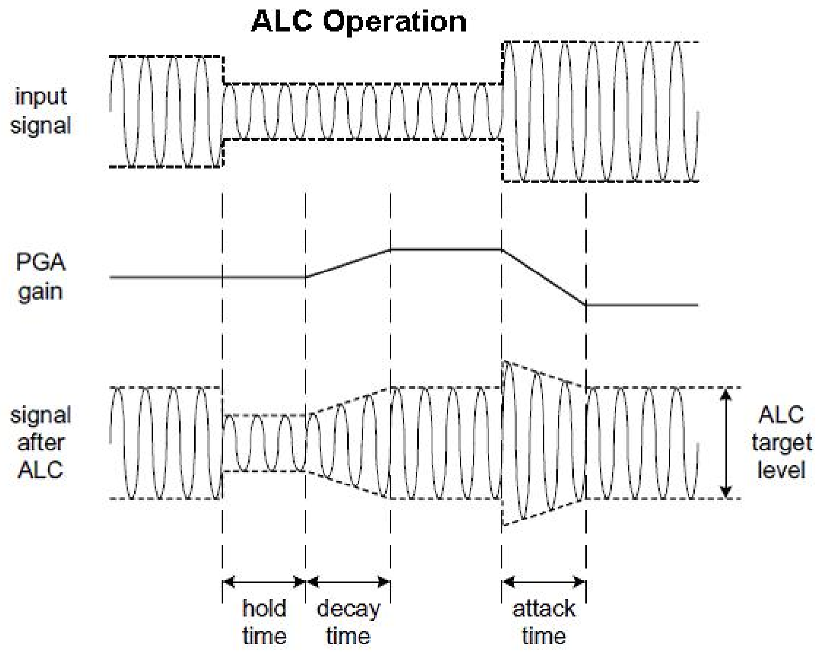
6.2.9 Register 17 – ADC Control 9, Default 1100 0000

Bit Name	Bit	Description
RADCVOL	7:0	Digital volume control attenuates the signal in 0.5 dB incremental from 0 to –96 dB. 00000000 – 0 dB 00000001 – -0.5 dB 00000010 – -1 dB … 11000000 – -96 dB (default)

9.3 AUTOMATIC LEVEL CONTROL (ALC)

In applications that offer a recording feature, ALC is often desirable to keep the recorded signal at a constant level. For example, if recording voice, the signal may vary a great deal depending on how loud the user speaks or how close to the mouth the microphone is held. This will result in a recorded signal that is difficult to listen to when played back.

The purpose of the ALC is to keep a constant output volume irrespective of the input signal level. This is achieved by continually adjusting the PGA gain so that the signal level at the ADC output remains constant.



Setting up the ALC to be optimal for each recorded source such as voice, classical music, pop music, etc. is quite a complex process. Recommended setups have been provided as a base to work from. The resultant effect is very subjective and may vary between applications. Some further modifications may be required to optimize the feature for a specific application but the recommended settings should offer suitable solutions in most cases.

There are some registers which can be use to control ALC. Please refer to the register listed below.

REGISTER ADDR (HEX)	REMARK	BIT[7]	BIT[6]	BIT[5]	BIT[4]	BIT[3]	BIT[2]	BIT[1]	BIT[0]	DEFAULT
Reg. 18	ALC Control 1	ALCSEL		MAXGAIN			MINGAIN			0011 1000
Reg. 19	ALC Control 2	ALCLVL				ALCHLD				1011 0000
Reg. 20	ALC Control 3	ALCDCY				ALCATK				0011 0010
Reg. 21	ALC Control 4	ALCMODE	ALCZC	TIME_OUT	WIN_SIZE					0000 0110
Reg. 22	ALC Control 5	NGTH					NGG		NGAT	0000 0000

9.3.1 CONTROL FIELDS

The ALC function in ES8388 CODEC is highly adaptable on account of the number of different parameters that may be individually set. The following paragraphs describe each of these parameters, and also some of the constraints or tradeoffs that determine the optimum setting for each. Different parameter values will be desirable to suit different types of audio signal. Personal preferences can also influence the choice of settings.

ALC Enable / ALC Level. The ALC function is enabled by setting the register field ALCSEL. ALCSEL=2'b00: ALC OFF; ALCSEL=2'b01: ALC Right Channel Enabled; ALCSEL=2'b10: ALC Left Channel Enabled; ALCSEL=2'b11: ALC Enabled. When enabled, the ALC output volume can be programmed using the ALCLVL register field. The range of ALC Level varies between 0dBFS and -16.5dBFS with 1.5dBFS per step. The maximum target level is always not above ADC full-scale level to help reduce the possibility of clipped signals. This level should be set as high as possible in order to achieve the best signal to noise performance, but not so high as to allow signal clipping to occur as the signal changes. The more erratic the signal level, the greater the required headroom between the ALC Target Level and the ADC full-scale level.

ALC Maximum Gain. An upper limit for the PGA gain is imposed by setting the register field MAXGAIN. The range of MAXGAIN varies between -6.5dBFS and 35.5dBFS with 6dBFS per step. The purpose of the maximum gain is to ensure the small input signals are accommodated and not excessively amplified by the ALC function. For example, if a recorded music track fades out; the Maximum Gain setting prevents ALC from destroying the effect by continually increasing the gain as the music signal fades. The Maximum Gain should be determined from the level of a quiet signal that the designer determines should be treated as a fading signal.

ALC Minimum Gain. A lower limit for the PGA gain is imposed by setting the register field MINGAIN. The range of MINGAIN varies between -12dBFS and 30dBFS with 6dBFS per step. The purpose of the minimum gain is to ensure that large input signals are permitted and not excessively attenuated by the ALC function. If the Minimum Gain is large, then the ALC will be

restricted in its ability to control the signal level and there is a greater possibility that it will be unable to prevent distortion of large signals. However, if the Minimum Gain is small, then a greater attenuation will be applied to large signals, which may undesirably limit the dynamic range of the processed signal. The Minimum Gain should be set as low as is possible, and certainly no greater than the gain that would be required to adjust the largest input signal down to the ALC target level.

ALC Hold Time. The time delay between the signal level detected below target level and the PGA gain beginning to ramp up, is controlled by register field ALCHLD. It can be set to zero, or can be programmed in power-of-two (2n) steps, e.g. 2.67ms, 5.33ms, 10.67ms etc. up to 1.36s. Note that the ALC Hold Time only applies to gain ramp-up; there is no delay before ramping the gain down when the signal level is above target. The ALC Hold Time is not active in Limiter Mode (see below).

The purpose of this delay is to ensure that the ALC is not over-responsive to a changing signal level. The Hold Time should be set according to the type of ALC response that is desired. A short Hold Time should be used if an immediate gain adjustment is required to a changing signal; this might be applicable to voice applications. A longer Hold Time should be used if the ALC gain adjustments are to be made more sparingly, thus responding only to long term signal level changes and preserving the original signal dynamics to a greater extent; this might be applicable to music containing a large dynamic range, as is frequently found in classical music.

ALC Decay Time. The time taken by the ALC for ramp up the PGA gain is controlled by register field ALCDCY. The ALC Decay Time is defined as a time per gain step. It can be programmed in power-of-two (2n) steps, e.g. 410us/step, 820us/step, 1.64ms/step, etc up to 420ms/step.

The Decay Time determines how rapidly the ALC will make adjustments in response to a fall in signal level. A short Decay Time should be used if a fast response is required to a changing signal. The Decay Time should not be so short as to cause rapid ALC response to a nominally constant signal level. For example, if the input signal is likely to have pauses or silences, the Decay Time should be set long enough to ensure that the ALC is prevented from making large adjustments to the gain during those durations.

ALC Attack Time. The time taken by the ALC for ramp down the PGA gain is controlled by register field ALCATK. The ALC Attack Time is defined as a time per gain step. It can be programmed in power-of-two (2n) steps, e.g. 104us/step, 208us/step, 416us/step, etc up to 106ms/6dB.

It is measured similarly to the Decay Time. The advantage of a short Attack Time is that it results in a fast response to an increased signal level. This in turn reduces the possibility of clipping. Many of the same considerations apply as for Hold Time and Decay Time. The Attack Time should be set in conjunction with the Decay Time. If the system requires fast response such as voice applications, a faster Attack/Decay Time may be needed. On the other hand, if a more steady input signal is anticipated, then a slower Attack/Decay Time may be most suitable.

ALC Mode. Two modes of operation are available via register bit ALCMODE. Normal ALC operation is selected by setting ALCMODE = 0. Limiter Mode is selected by setting ALCMODE = 1. In Limiter mode, the ALC Maximum Gain is set equal to the PGA setting at the time that limiter mode is entered. In this mode, the signal level may be reduced to prevent overload, but may not be increased above the initial PGA gain setting - the register field MAXGAIN are not used when Limiter Mode is selected. In Limiter mode, the gain control circuit runs approximately 4 times faster to allow quick reduction of high signal levels and quick increment of low signal levels.

Don't use limiter mode during microphone recording.

Peak Limiter. To avoid clipping when a large signal is applied just after a period where the PGA gain has been ramped up (eg. after a period of quiet), the ALC circuit includes a Peak Limiter function. If the input signal after PGA gain exceeding the Peak threshold (fixed at 1.5dB below full scale), the PGA gain is ramped down at the maximum attack rate (as when ALCATK = 0000), until the signal level falls below the Peak threshold. This function is always enabled whenever the ALC is enabled.

Sample Rate. The ALC Hold, Decay and Attack times will vary slightly depending on the sample rate used. For example, when sampling at 48kHz, the ALC Hold, Decay and Attack times are equal to the values in registers' definition. If the actual sample rate was 44.1kHz, then the ALC Hold, Decay and Attack times would be scaled down by 44.1/48.

ALC Zero Cross/Time out. The register field ALCZC is used to control whether PGA gain updates are timed to occur at the

zero-cross points of the input signal. Enabling this feature ensures that pops and clicks arising from the PGA gain adjustments are minimized, but this feature may also result in a faster ALC response. If the PGA gain steps are small enough to eliminate the need for Zero-Cross detection, the Zero-Cross option is not required.

A timeout is provided to ensure that the gain may still be updated if a zero-cross has not occurred within a fixed time. The timeout is enabled via register setting TIME_OUT, it is not automatically enabled.

Noise Gate Enable / Noise Gate Threshold / Noise Gate Type. The Noise Gate function is enabled by setting the register field NGAT. When enabled, the Noise Gate Threshold can be programmed using the NGTH register field. The range of Noise Gate Threshold varies between -76.5dBFS and -30dBFS with 1.5dBFS per step; this threshold is the input signal level below which the PGA gain will either be muted or be held constant. the Noise Gate type (Mute or Hold Gain) can be set by register field NGG.

9.3.2 RECOMMENDED SETTINGS FOR ALC

Recommended settings are provided below for a number of typical portable recording applications. These include voice recording and music recording. A generic setting is also provided, which aims to cater for the widest possible range of sounds.

It is important to note that these are suggested initial values only, as a starting point from which to derive the best settings for a particular circuit application. The quoted settings should give adequate performance in many cases, but it may be possible to improve the ALC performance through further adjustment of these settings.

For voice recording, a fast ALC response is desirable in order to quickly compensate for different peoples voices, movement relative to the microphone.

For music recording, the fast response is not recommended as it is likely to result in clipping in response to any sudden changes in the music signal level. A reduction in the maximum gain setting may help to avoid clipping when the music level increases after a quiet period and to restrict the extent of the ALC adjustments. This may not be desirable in all music applications and is therefore not shown in the recommended settings. It is one of the many adjustments that the user should consider when optimizing for a known operational environment.

For generic recording, the ALC must attempt to accommodate all types of sounds, a compromise setting must be found. For an ALC response that is tolerant to impulses such as handclaps, it is recommended that the ALC Attack time should not be set too fast and the ALC Decay time should not be set too slow. The minimum and maximum gain settings could be adjusted to restrict the extent of the ALC control if desired. The combination of settings should allow the ALC to respond quickly to changes in signal level and to impulse-type sounds, but also to minimize gain pumping caused by the associated level changes.

The settings in following table are the recommended settings for voice, music and generic (hand clap).

REGISTER BIT	RECOMMENDED VALUE		
	VOICE (Microphone Recording)	MUSIC	GENERIC(handclap)
ALCSEL[1:0]	11 (Stereo)	11 (Stereo)	11 (Stereo)
ALCLVL[3:0]	1100 (-4.5dB)	0011 (-12dB)	0011 (-12dB)
MAXGAIN[2:0]	101 (+23.5dB)	111 (+35.5dB)	111 (+35.5dB)
MINGAIN[2:0]	010 (0dB)	000 (-12dB)	000 (-12dB)
ALCHLD[3:0]	0000 (0ms)	0000 (0ms)	0000 (0ms)
ALCDCY[3:0]	0001 (820us/step)	1010 (420ms/step)	0101 (13.1ms/step)
ALCATK[3:0]	0010 (416us/step)	0110 (6.66ms/step)	0111 (13.3ms/step)
ALCMODE	0 (ALC)	0 (ALC)	0 (ALC)
ALCZC	0 (ZC off)	0(ZC off)	0 (ZC off)
TIME_OUT	0 (not enable)	0(not enable)	0(not enable)
NGAT	1 (enabled)	1 (enabled)	1 (enabled)
NGTH[4:0]	11000 (-40.5dB)	01011(-60dB)	10001 (-51dB)
NGG[1:0]	01 (mute ADC)	00 or 10 (hold gain)	00 or 10 (hold gain)
WIN_SIZE[4:0] MicAmpl Gain Reg. 0x09	00110(default) 0x77 (+21dB)	00110(default)	00110(default)

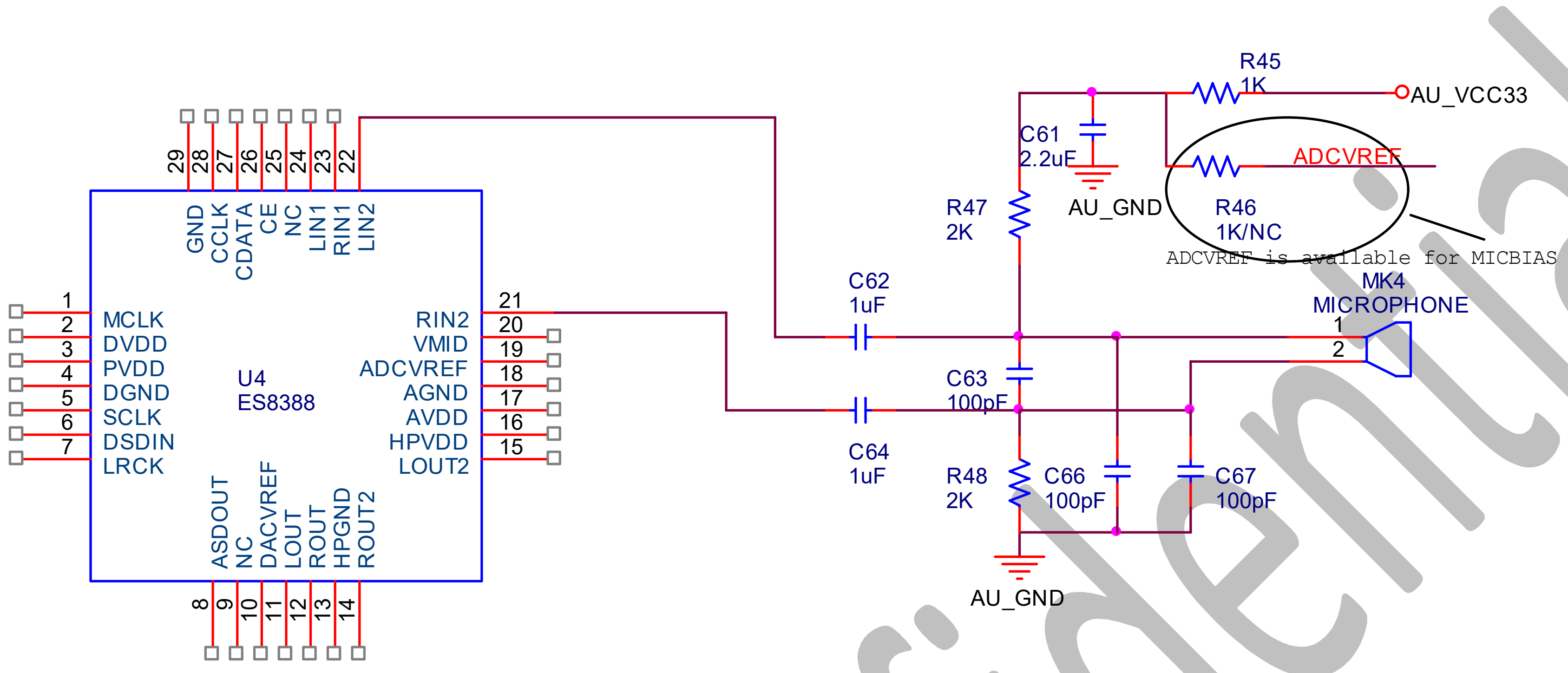
9.4 MICROPHONE INPUT CIRCUIT AND THE SAMPLE CODE FOR RECORDING

In recording mode, the analog input pins can be used as single ended input or differential input.

Some microphone input circuits (single ended input or differential input) and sample code for recording are listed below. In microphone recording mode, ALC should be enabled to boost the microphone input signal.

Please note that the fully-differential input circuit and pseudo-differential input circuit are recommended to ES8388 microphone recording.

9.4.1 FULLY-DIFFERENTIAL MICROPHONE INPUT CIRCUIT AND SAMPLE CODE



Notes:

1. On PCB layout, these wires which be marked as red must be paralleled each other.

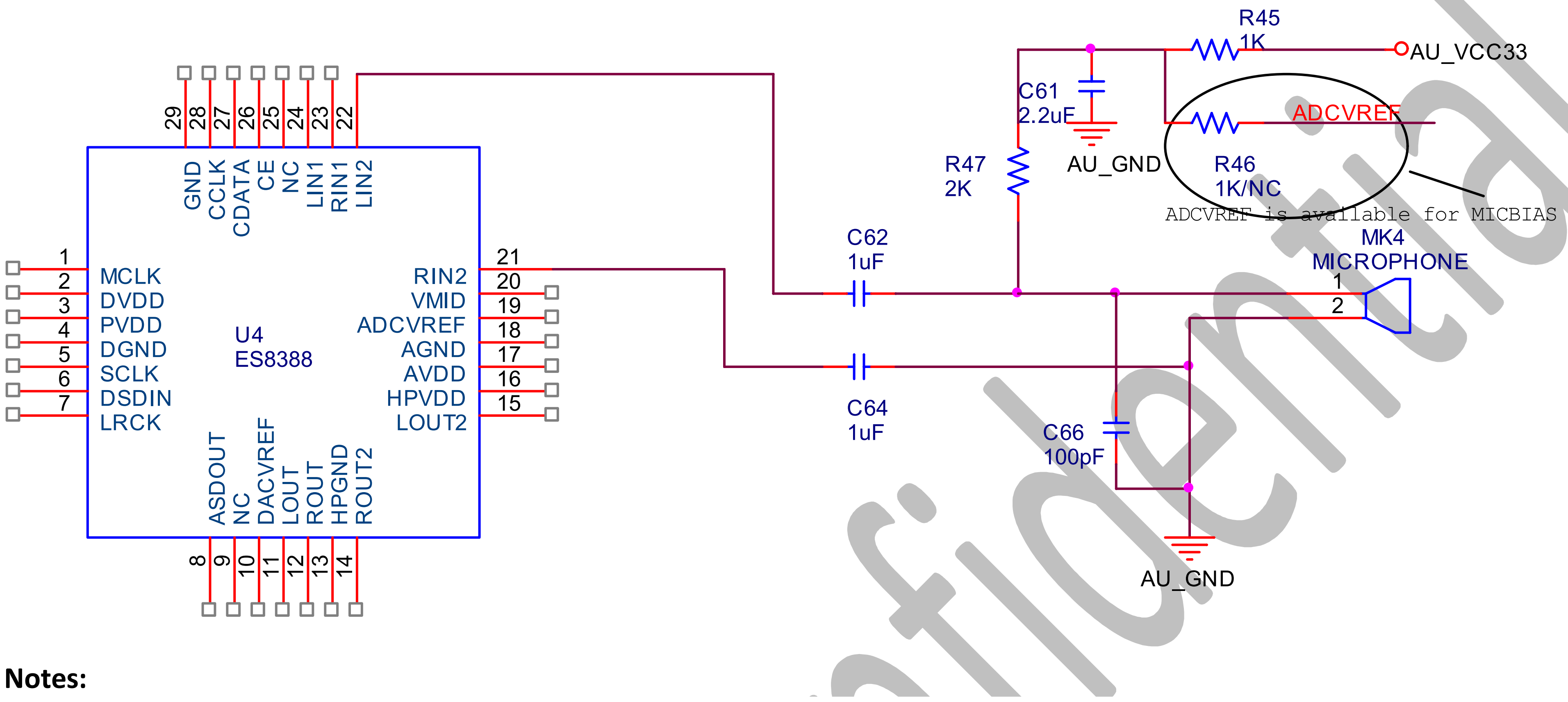
2. On PCB layout, the components such as R45, R46, R47, R48, C61, C62, C63, C64, C66 and C67 must be located as close to Microphone as possible.

The sample code for microphone differential input:

```
 es8388_write( 0x02, 0xF3);   // Reg0x02 = 0xF3, Stop STM, DLL, and digital block
es8388_write(0x08, 0x00);     // Reg 0x08 = 0x00, ES8388 in I2S slave mode
//es8388_write( 0x08 0x80);   // Reg 0x08 = 0x80 (ES8388 in I2S master mode)
 es8388_write( 0x2B, 0x80);   // Reg 0x2B = 0x80 (Set ADC and DAC have the same LRCK)
 es8388_write( 0x00, 0x05);   // Reg 0x00 = 0x05 (start up reference)
 es8388_write( 0x01, 0x40);   // Reg 0x01 = 0x40 (start up reference)
 es8388_write( 0x03, 0x00);   // Reg 0x03 = 0x00 (Power on ADC and LIN/RIN input)
 es8388_write( 0x09 0x77);     // Reg 0x09 = 0x77 (MicBoost PGA=+21Db)
 es8388_write( 0x0A 0xF0);     // Reg 0x0A = 0xF0 (differential input)
 es8388_write( 0x0B 0x02);     // Reg 0x0B = 0x02 (Select LIN1and RIN1 as differential input pairs)
  // es8388_write( 0x0B 0x82);     // Reg 0x0B = 0x82 (Select LIN2and RIN2 as differential input pairs)
 es8388_write( 0x0C 0x40);     // Reg 0x0C = 0x00 (I2S – 24bits, Ldata = LADC, Rdata = RADC)
 es8388_write( 0x0D 0x02);     // Reg 0x0D = 0x02 (MCLK/LRCK = 256)


es8388_write( 0x10 0x00);     // Reg 0x10= 0x00 (LADC volume = 0dB)
es8388_write( 0x11 0x00);     // Reg 0x11= 0x00 (RADC volume = 0dB)
es8388_write(0x12, 0xe2);     // Reg 0x12 = 0xe2 (ALC enable, PGA Max. Gain=23.5dB, Min. Gain=0dB)
es8388_write( 0x13, 0xa0);   // Reg 0x13 = 0xc0 (ALC Target=-4.5dB, ALC Hold time =0 mS)
```

```
es8388_write( 0x14, 0x12);    // Reg 0x14 = 0x12(Decay time =820uS , Attack time = 416 uS)

es8388_write( 0x15, 0x06);    // Reg 0x15 = 0x06(ALC mode)

es8388_write( 0x16, 0xc3);    // Reg 0x16 = 0xc3(nose gate = -40.5dB, NGG = 0x01(mute ADC))

es8388_write( 0x02, 0x55);    // Reg 0x16 = 0x55 (Start up DLL, STM and Digital block for recording)
```

9.4.2 PSEUDO-DIFFERENTIAL MICROPHONE INPUT CIRCUIT AND SAMPLE CODE



Notes:

1. On PCB layout, these wires which be marked as red must be paralleled each other.

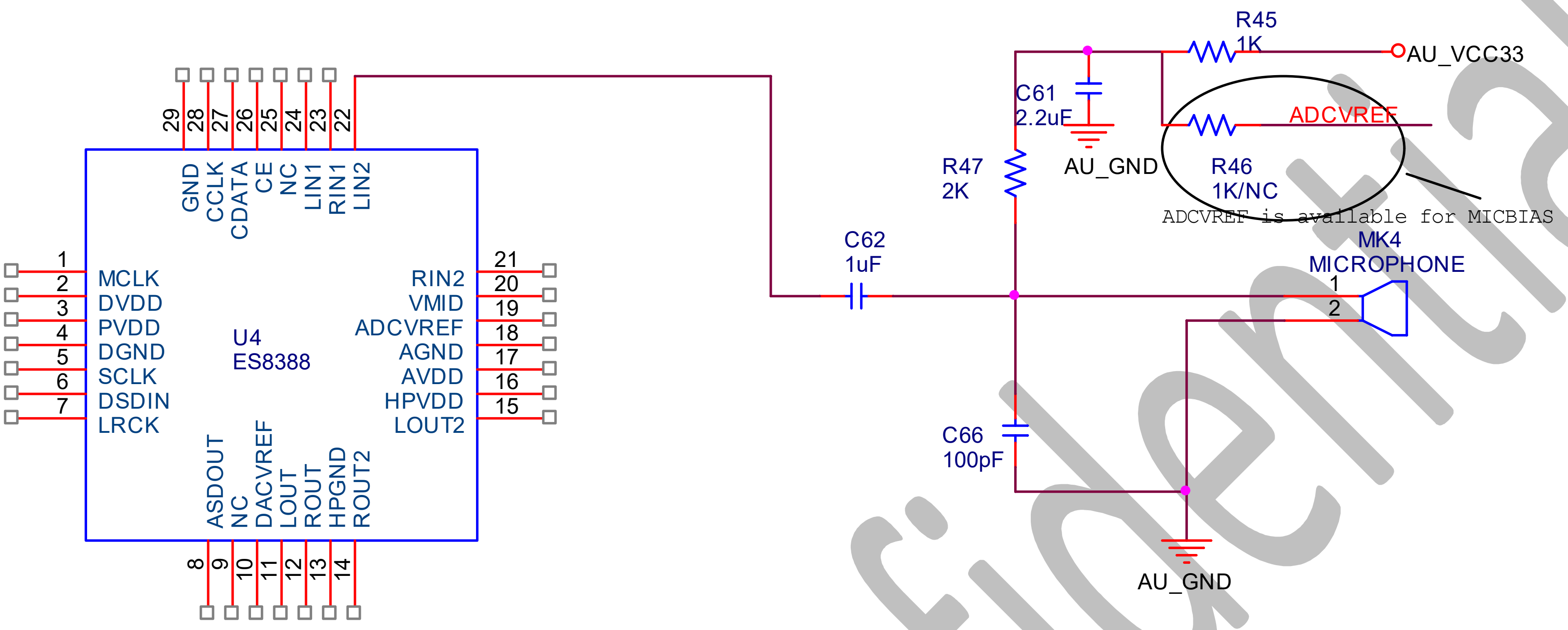
2. On PCB layout, the components such as R45, R46, R47, C61, C62, C64 and C66 must be located as close to Microphone as possible.

The sample code for microphone pseudo differential input:

```
es8388_write( 0x02, 0xF3);    // Reg0x02 = 0xF3, Stop STM, DLL, and digital block

es8388_write(0x08, 0x00);     // Reg 0x08 = 0x00, ES8388 in I2S slave mode

//es8388_write( 0x08 0x80);    // Reg 0x08 = 0x80 (ES8388 in I2S master mode)

es8388_write( 0x2B, 0x80);    // Reg 0x2B = 0x80 (Set ADC and DAC have the same LRCK)

es8388_write( 0x00, 0x05);    // Reg 0x00 = 0x05 (start up reference)

es8388_write( 0x01, 0x40);    // Reg 0x01 = 0x40 (start up reference)

es8388_write( 0x03, 0x00);    // Reg 0x03 = 0x00 (Power on ADC and LIN/RIN input)

es8388_write( 0x09 0x77);     // Reg 0x09 = 0x77 (MicBoost PGA=+21Db)

es8388_write( 0x0A 0xF0);     // Reg 0x0A = 0xF0 (differential input)

es8388_write( 0x0B 0x02);     // Reg 0x0B = 0x02 (Select LIN1and RIN1 as differential input pairs)

// es8388_write( 0x0B 0x82);    // Reg 0x0B = 0x82 (Select LIN2and RIN2 as differential input pairs)

es8388_write( 0x0C 0x40);     // Reg 0x0C = 0x00 (I2S – 24bits, Ldata = LADC, Rdata = RADC)

es8388_write( 0x0D 0x02);     // Reg 0x0D = 0x02 (MCLK/LRCK = 256)


es8388_write( 0x10 0x00);     // Reg 0x10= 0x00 (LADC volume = 0dB)

es8388_write( 0x11 0x00);     // Reg 0x11= 0x00 (RADC volume = 0dB)

es8388_write(0x12, 0xe2);     // Reg 0x12 = 0xe2 (ALC enable, PGA Max. Gain=23.5dB, Min. Gain=0dB)
```

```
es8388_write( 0x13, 0xa0);    // Reg 0x13 = 0xc0 (ALC Target=-4.5dB, ALC Hold time =0 mS)

es8388_write( 0x14, 0x12);    // Reg 0x14 = 0x12(Decay time =820uS , Attack time = 416 uS)

es8388_write( 0x15, 0x06);    // Reg 0x15 = 0x06(ALC mode)

es8388_write( 0x16, 0xc3);   // Reg 0x16 = 0xc3(nose gate = -40.5dB, NGG = 0x01(mute ADC))

es8388_write( 0x02, 0x55);    // Reg 0x16 = 0x55 (Start up DLL, STM and Digital block for recording)
```

9.4.3 SINGLE ENDED MICROPHONE INPUT CIRCUIT AND SAMPLE CODE



Notes:

1. On PCB layout, the components such as R45, R46, R47, C61, C62 and C66 must be located as close to Microphone as possible.

2. In customer application, the microphone single ended input circuit must not be recommended for ES8388 microphone recording.

The sample code for microphone single ended input:

```
es8388_write( 0x02, 0xF3);    // Reg0x02 = 0xF3, Stop STM, DLL, and digital block
es8388_write(0x08, 0x00);      // Reg 0x08 = 0x00, ES8388 in I2S slave mode
//es8388_write( 0x08 0x80);    // Reg 0x08 = 0x80 (ES8388 in I2S master mode)
es8388_write( 0x2B, 0x80);    // Reg 0x2B = 0x80 (Set ADC and DAC have the same LRCK)
es8388_write( 0x00, 0x05);    // Reg 0x00 = 0x05 (start up reference)
es8388_write( 0x01, 0x40);    // Reg 0x01 = 0x40 (start up reference)
es8388_write( 0x03, 0x00);    // Reg 0x03 = 0x00 (Power on ADC and LIN/RIN input)
es8388_write( 0x09 0x77);      // Reg 0x09 = 0x77 (MicBoost PGA=+21Db)
es8388_write( 0x0A 0x00);      // Reg 0x0A = 0xF0 (Lin1 and RIN1 used as single ended input)
//es8388_write( 0x0A 0x50);      // Reg 0x0A = 0xF0 (Lin2 and RIN2 used as single ended input)
es8388_write( 0x0C 0x40);      // Reg 0x0C = 0x00 (I2S – 24bits, Ldata = LADC, Rdata = RADC)
es8388_write( 0x0D 0x02);      // Reg 0x0D = 0x02 (MCLK/LRCK = 256)
es8388_write( 0x10 0x00);      // Reg 0x10= 0x00 (LADC volume = 0dB)
es8388_write( 0x11 0x00);      // Reg 0x11= 0x00 (RADC volume = 0dB)
es8388_write(0x12, 0xe2);      // Reg 0x12 = 0xe2 (ALC enable, PGA Max. Gain=23.5dB, Min. Gain=0dB)
es8388_write( 0x13, 0xa0);    // Reg 0x13 = 0xc0 (ALC Target=-4.5dB, ALC Hold time =0 mS)
es8388_write( 0x14, 0x12);    // Reg 0x14 = 0x12(Decay time =820uS , Attack time = 416 uS)
es8388_write( 0x15, 0x06);    // Reg 0x15 = 0x06(ALC mode)
es8388_write( 0x16, 0xc3);   // Reg 0x16 = 0xc3(nose gate = -40.5dB, NGG = 0x01(mute ADC))
es8388_write( 0x02, 0x55);    // Reg 0x16 = 0x55 (Start up DLL, STM and Digital block for recording)
```

10 OUTPUT SIGNAL PATH

ES8388 output signal path consist of digital filters, DACs, Analog mixers and Output drivers. The digital filters and DACs are enabled when ES8388 is in 'play back mode' or 'record and play back mode'. The mixers and output drivers can be separately enabled by individual control bits. Thus it is possible to utilize the analog mixing and amplification provided by ES8388, irrespective of whether the DACs are running or not.

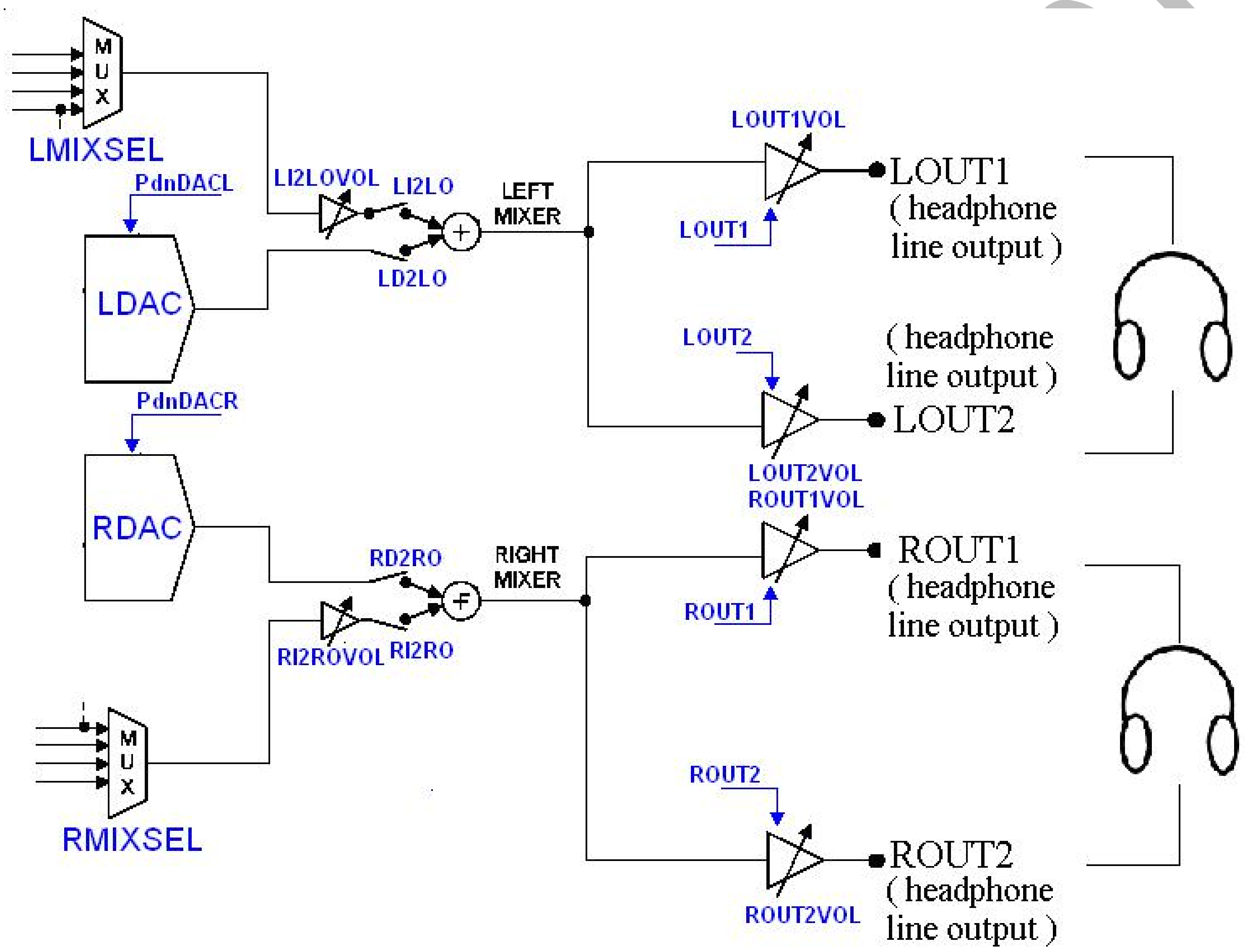
ES8388 receives digital input data on DSDIN pin. The digital filter block process the data to provide the following functions:

- Digital volume control
- Equalizer and stereo enhancement
- Sigma-Delta Modulation

Two high performance sigma-delta audio DACs convert the digital input data into two analog signals(left and right). These can be mixed with analog input signals from LIN1/2 and RIN1/2 pins, and the mixed is fed to the output drivers,LOUT1/ROUT1 and LOUT2/ROUT2.

- LOUT1/ROUT1: can drive 16Ω or 32Ω stereo headphone or stereo line output
- LOUT2/ROUT2: can drive 16Ω or 32Ω stereo headphone or stereo line output

10.1 THE OUTPUT SIGNAL PATHS AND THE CONTROL REGISTER



PdnDACL and PdnDACR control bits in register 4 are used to power up / power down Left and Right channel DAC.

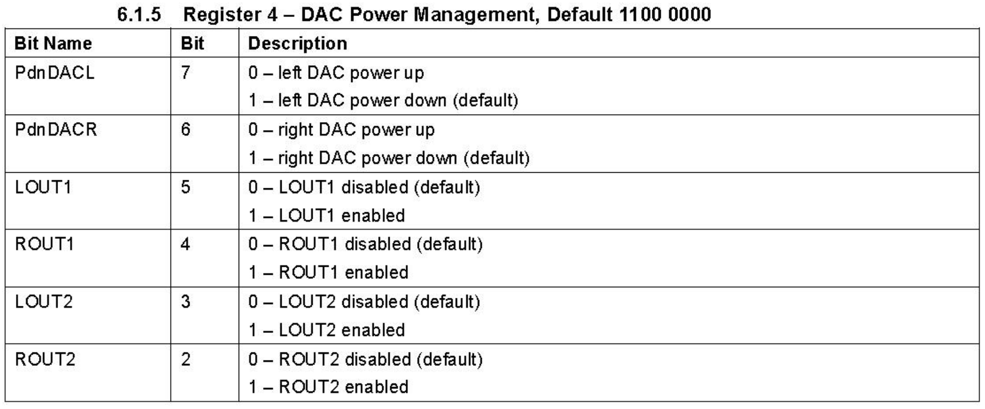
LOUT1, ROUT1, LOUT2 and ROUT2 control bits in register 4 are used to power up / power down output drivers, LOUT1, ROUT1, LOUT2 and ROUT2.

The signal volume on LOUT1, ROUT1, LOUT2 and ROUT2 can be independently adjusted under software control by writing LOUT1VOL, ROUT1VOL, LOUT2VOL and ROUT2VOL, respectively. The output driver volume is adjustable from -45dB to +4.5dB in 1.5dB steps. Note that the volume over 0dB may cause clipping if signal is large.

Left and right mixers are used to mix the DAC's output and the analog input signal which be selected by LMIXSEL and RMIXSEL.

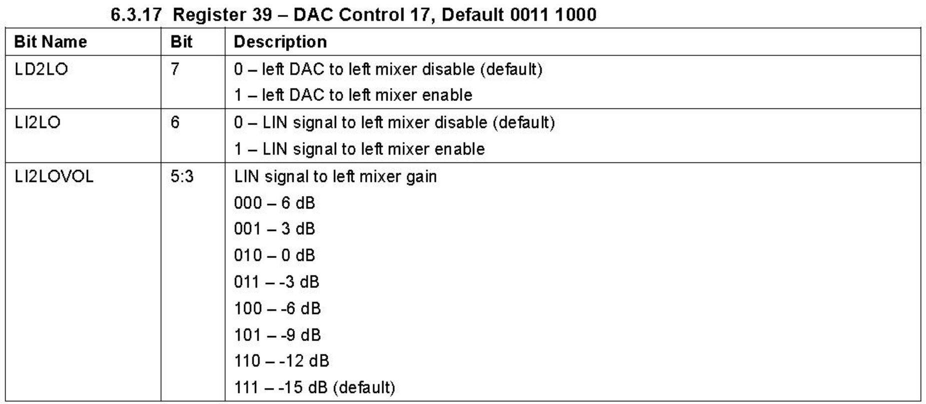
If LI2LO control bits in register 39 is set to '1',the analog input signal which be selected by LMIXSEL should be mixed in left mixer. If RI2RO control bits in register 42 is set to '1',the analog input signal which be selected by RMIXSEL should be mixed in right mixer.

The mixed signal can be controlled with Mixer PGAs (LI2LOVOL and RI2ROVOL). The gain of Mixer PGA is adjustable from -15dB to +6dB in 3dB steps. The left mixed is fed to LOUT1 and LOUT2 output drivers, and the right mixed is fed to ROUT1 and ROUT2 output drivers.



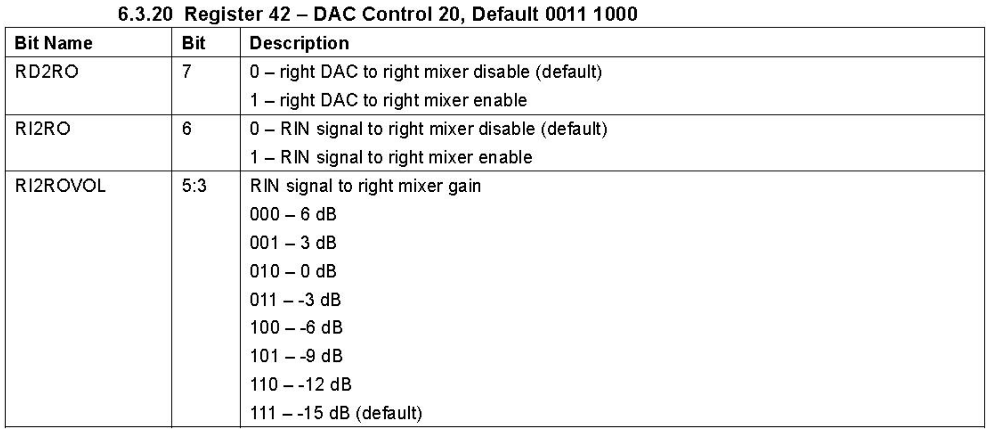
6.1.5 Register 4 – DAC Power Management, Default 1100 0000

Bit Name	Bit	Description
PdnDACL	7	0 – left DAC power up 1 – left DAC power down (default)
PdnDACR	6	0 – right DAC power up 1 – right DAC power down (default)
LOUT1	5	0 – LOUT1 disabled (default) 1 – LOUT1 enabled
ROUT1	4	0 – ROUT1 disabled (default) 1 – ROUT1 enabled
LOUT2	3	0 – LOUT2 disabled (default) 1 – LOUT2 enabled
ROUT2	2	0 – ROUT2 disabled (default) 1 – ROUT2 enabled



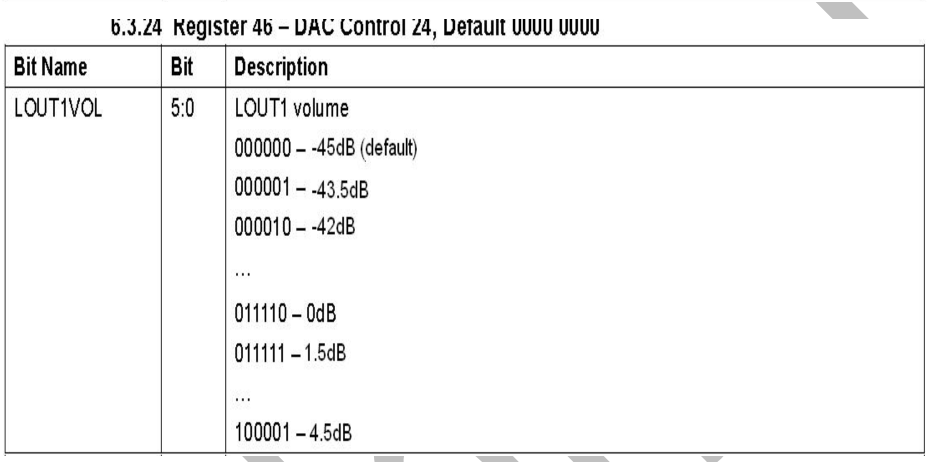
6.3.17 Register 39 – DAC Control 17, Default 0011 1000

Bit Name	Bit	Description
LD2LO	7	0 – left DAC to left mixer disable (default) 1 – left DAC to left mixer enable
LI2LO	6	0 – LIN signal to left mixer disable (default) 1 – LIN signal to left mixer enable
LI2LOVOL	5:3	LIN signal to left mixer gain 000 – 6 dB 001 – 3 dB 010 – 0 dB 011 – -3 dB 100 – -6 dB 101 – -9 dB 110 – -12 dB 111 – -15 dB (default)



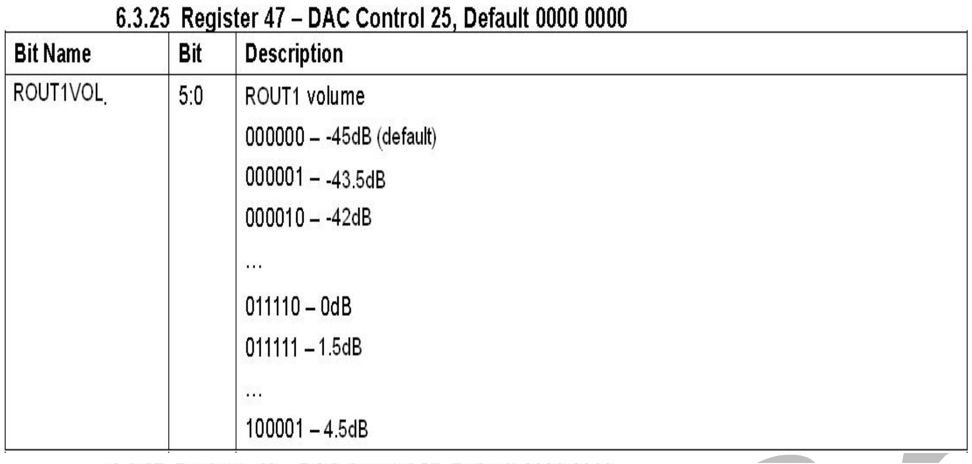
6.3.20 Register 42 – DAC Control 20, Default 0011 1000

Bit Name	Bit	Description
RD2RO	7	0 – right DAC to right mixer disable (default) 1 – right DAC to right mixer enable
RI2RO	6	0 – RIN signal to right mixer disable (default) 1 – RIN signal to right mixer enable
RI2ROVOL	5:3	RIN signal to right mixer gain 000 – 6 dB 001 – 3 dB 010 – 0 dB 011 – -3 dB 100 – -6 dB 101 – -9 dB 110 – -12 dB 111 – -15 dB (default)



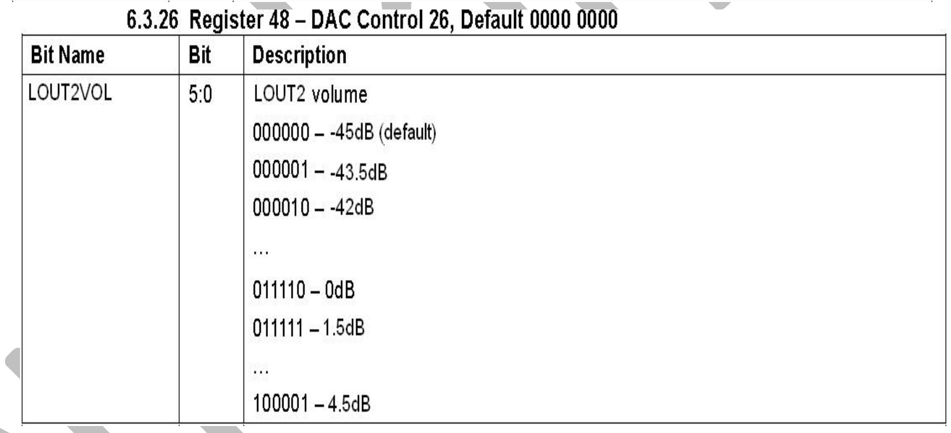
6.3.24 Register 46 – DAC Control 24, Default 0000 0000

Bit Name	Bit	Description
LOUT1VOL	5:0	LOUT1 volume 000000 – -45dB (default) 000001 – -43.5dB 000010 – -42dB … 011110 – 0dB 011111 – 1.5dB … 100001 – 4.5dB



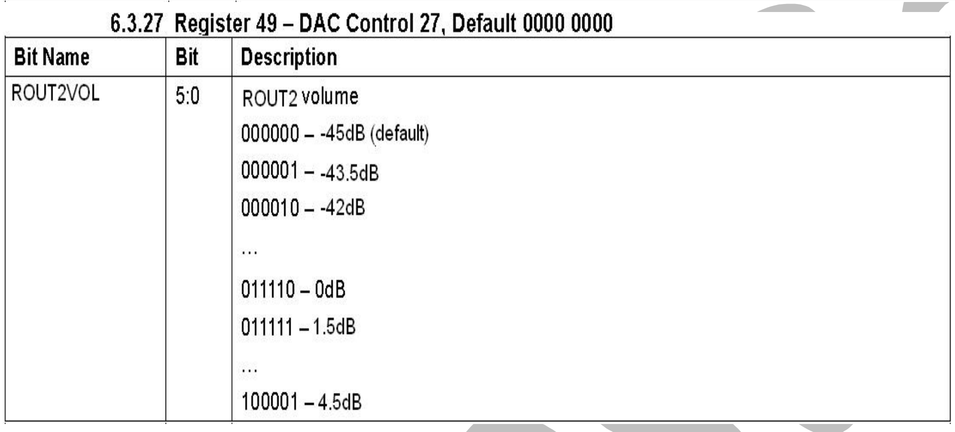
6.3.25 Register 47 – DAC Control 25, Default 0000 0000

Bit Name	Bit	Description
ROUT1VOL	5:0	ROUT1 volume 000000 – -45dB (default) 000001 – -43.5dB 000010 – -42dB … 011110 – 0dB 011111 – 1.5dB … 100001 – 4.5dB



6.3.26 Register 48 – DAC Control 26, Default 0000 0000

Bit Name	Bit	Description
LOUT2VOL	5:0	LOUT2 volume 000000 – -45dB (default) 000001 – -43.5dB 000010 – -42dB … 011110 – 0dB 011111 – 1.5dB … 100001 – 4.5dB



6.3.27 Register 49 – DAC Control 27, Default 0000 0000

Bit Name	Bit	Description
ROUT2VOL	5:0	ROUT2 volume 000000 – -45dB (default) 000001 – -43.5dB 000010 – -42dB … 011110 – 0dB 011111 – 1.5dB … 100001 – 4.5dB

10.2 THE DAC CONTROL REGISTER

PdnDACL and PdnDACR control bits in register 4 are used to power up / power down Left and Right channel DAC. Please refer to section 9.2.

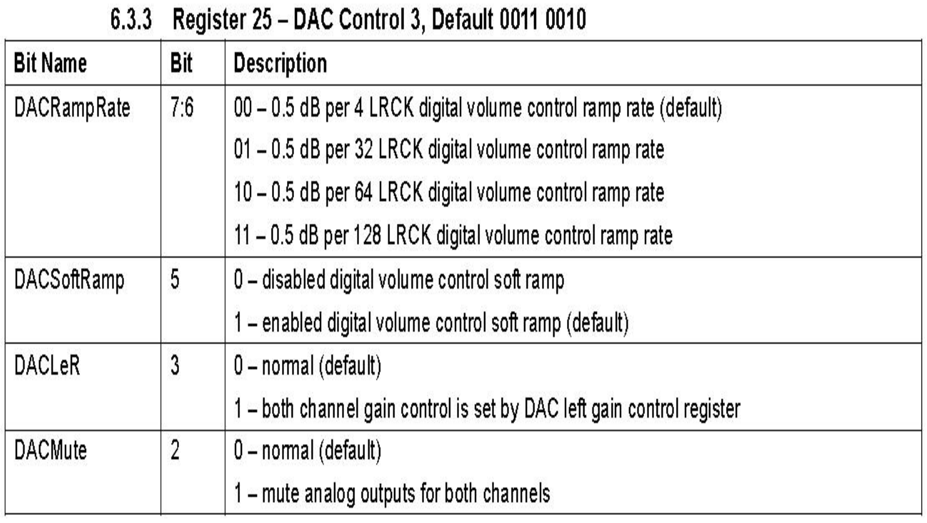
ES8388 DAC's digital volume can be adjustable from -96dB to 0dB in 0.5dB steps. LDACVOL and RDACVOL is used to independently control the left an right DAC volume. If DACLeR control bit in register 25 is set to '1', the volume of left and right DAC can adjust synchronously by adjusting left DAC volume. DACs should be mute if DACMute control bit in register 25 is set to '1'.

ZeroL and ZeroR control bits in register 29 are used to set the left and right channel DACs output all zero. This operation is equivalent to mute the DAC or set the DAC digital volume to -96dB.

DAC_invL control bit in register 28 are used to invert the left channel DAC ouput, and DAC_invR control bit in register 28 is used for right channel DAC.

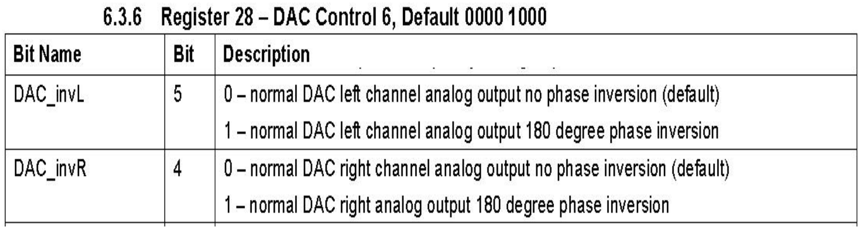
DACSoft Ramp control bit in register 25 is use to fade in and fade out. DACRampRate control bits are used to set the soft ramp rate.

Mono control bit is used to set the DAC in stereo output mode or mono output mode.



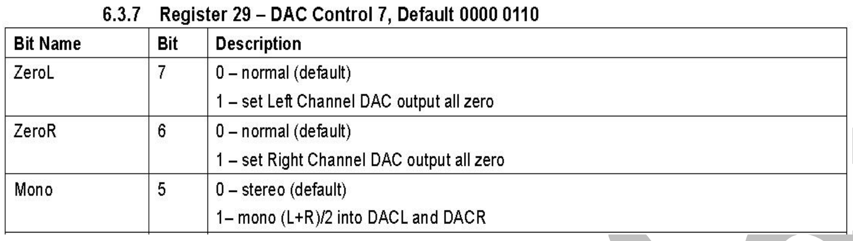
6.3.3 Register 25 – DAC Control 3, Default 0011 0010

Bit Name	Bit	Description
DACRampRate	7:6	00 – 0.5 dB per 4 LRCK digital volume control ramp rate (default) 01 – 0.5 dB per 32 LRCK digital volume control ramp rate 10 – 0.5 dB per 64 LRCK digital volume control ramp rate 11 – 0.5 dB per 128 LRCK digital volume control ramp rate
DACSoftRamp	5	0 – disabled digital volume control soft ramp 1 – enabled digital volume control soft ramp (default)
DACLeR	3	0 – normal (default) 1 – both channel gain control is set by DAC left gain control register
DACMute	2	0 – normal (default) 1 – mute analog outputs for both channels



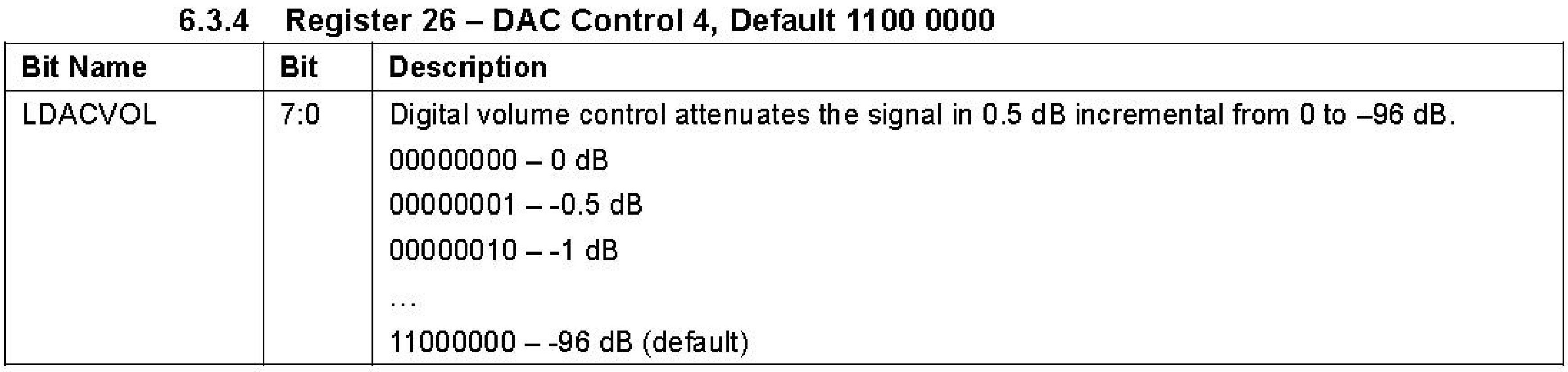
6.3.6 Register 28 – DAC Control 6, Default 0000 1000

Bit Name	Bit	Description
DAC_invL	5	0 – normal DAC left channel analog output no phase inversion (default) 1 – normal DAC left channel analog output 180 degree phase inversion
DAC_invR	4	0 – normal DAC right channel analog output no phase inversion (default) 1 – normal DAC right analog output 180 degree phase inversion



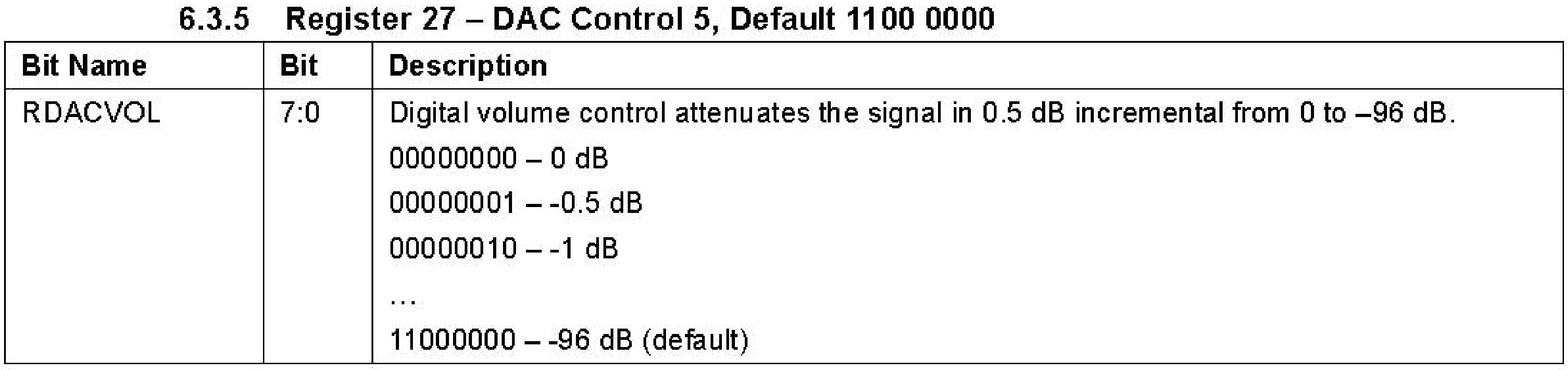
6.3.7 Register 29 – DAC Control 7, Default 0000 0110

Bit Name	Bit	Description
ZeroL	7	0 – normal (default) 1 – set Left Channel DAC output all zero
ZeroR	6	0 – normal (default) 1 – set Right Channel DAC output all zero
Mono	5	0 – stereo (default) 1– mono (L+R)/2 into DACL and DACR



6.3.4 Register 26 – DAC Control 4, Default 1100 0000

Bit Name	Bit	Description
LDACVOL	7:0	Digital volume control attenuates the signal in 0.5 dB incremental from 0 to –96 dB. 00000000 – 0 dB 00000001 – -0.5 dB 00000010 – -1 dB … 11000000 – -96 dB (default)



6.3.5 Register 27 – DAC Control 5, Default 1100 0000

Bit Name	Bit	Description
RDACVOL	7:0	Digital volume control attenuates the signal in 0.5 dB incremental from 0 to –96 dB. 00000000 – 0 dB 00000001 – -0.5 dB 00000010 – -1 dB … 11000000 – -96 dB (default)

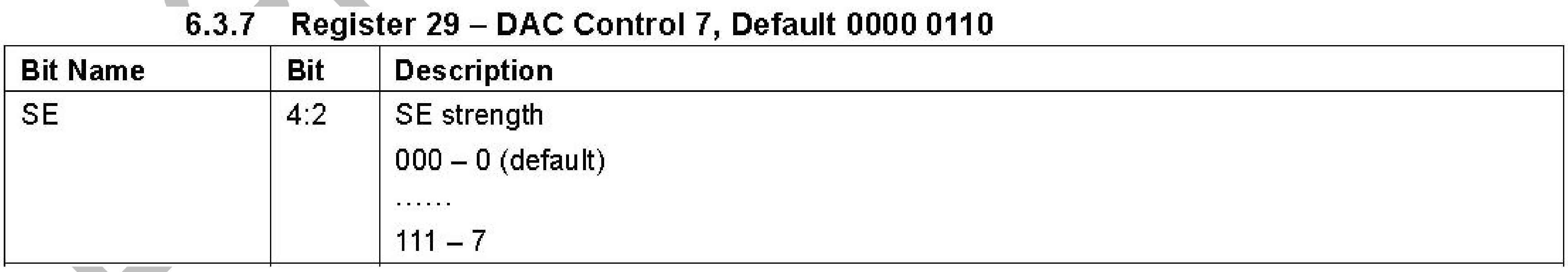
10.3 EQUALIZER AND STEREO ENHANCEMENT

ES8388 provides equalizer and stereo enhancement function in play back mode. These can't be used in Bypass mode.

For equalizer, only 2 band equalizer is used. It can do bass or treble operation, but can't do bass and treble operation at the same time. Everest Semiconductor Co., Ltd will provide equalizer calculator to help user to utilize this equalizer.

The equalizer register address is located from register 30 to register 37.

The SE control bits in register 29 are used to set the stereo strength. There are 8 levels stereo strength from 0 to 7. ES8388 will get the strongest stereo effect if SE equal to 7. There is not any stereo enhancement if SE equal to 0.



6.3.7 Register 29 – DAC Control 7, Default 0000 0110

Bit Name	Bit	Description
SE	4:2	SE strength 000 – 0 (default) …… 111 – 7

Reg. 30			Shelving_a[29:24]
Reg. 31	Shelving_a[23:16]		
Reg. 32	Shelving_a[15:8]		
Reg. 33	Shelving_a[7:0]		
Reg. 34			Shelving_b[29:24]
Reg. 35	Shelving_b[23:16]		
Reg. 36	Shelving_b[15:8]		
Reg. 37	Shelving_b[7:0]		

11 REGISTER CONFIGURATION FOR ES8388

The Register configuration include start up codec mode, start up recording mode, start up play back mode, start up by pass mode, power down (to standby mode), resume from standby mode, etc.

11.1 THE SEQUENCE FOR START UP CODEC

```
void ES8388_CODEC_INI(void)
{
 unsigned int i;
 WriteReg(0x01, 0x58);      //Power down whole chip analog
 WriteReg(0x01, 0x50);      //power up whole chip analog
 WriteReg(0x02, 0xF3);      //stop STM and DLL, power down DAC&ADC vref
 WriteReg(0x02, 0xF0);      //power up DAC&ADC vref
 WriteReg(0x2B, 0x80);      //set internal ADC and DAC use the same LRCK clock, dac LRCK as internal LRCK
 WriteReg(0x00, 0x36);      //ADC clock is same as DAC. Use DAC MCLK as internal MCLK

 #if defined(CODEC_I2S_MASTER_MODE)
     WriteReg(0x08, 0x80);   //CODEC IN I2S MASTER MODE
 #elif defined(CODEC_I2S_SLAVE_MODE)
     WriteReg(0x08, 0x00);   //CODEC IN I2S SLAVE MODE
 #endif

 WriteReg(0x04, 0x00);   //power up DAC
 WriteReg(0x05, 0x00);   //low power setting
 WriteReg(0x06, 0xC3);

 #if defined(SINGLE_ENDED_MIC)     //Single ended microphone input
     WriteReg(0x0A, 0x00);   //select LIN1&RIN1 as Microphone
     //WriteReg(0x0A, 0x50); // select LIN2&RIN2 as Microphone
#elif defined(DIFF_ENDED_MIC)       //differential microphone input
      WriteReg(0x0A, 0xF0);   //set differential input mode
      WriteReg(0x0B, 0x02); // select LIN1&RIN1 as Microphone

 //WriteReg(0x0B, 0x82); //elect LIN2&RIN2 as Microphone
 #endif

 WriteReg(0x0C, 0x00);   //ADC I2S-24BIT
 WriteReg(0x17, 0x00);     //DAC I2S-24BIT

 WriteReg(0x0D, 0x02);   //ADCLRCK = MCLK/256
 WriteReg(0x18, 0x02);     //DACLRCK = MCLK/256

 WriteReg(0x10, 0x00);   //ADC VOLUME = 0DB
 WriteReg(0x11, 0x00);
 WriteReg(0x1A, 0x00);   //DAC VOLUME = 0DB
 WriteReg(0x1B, 0x00);

 WriteReg(0x09, 0x88);     //MIC PGA =24DB
 WriteReg(0x12, 0xE2);     //MIC ALC SETTING
 WriteReg(0x13, 0xC0);
 WriteReg(0x14, 0x12);
 WriteReg(0x15, 0x06);
 WriteReg(0x16, 0xC3);

WriteReg(0x27, 0xB8);       //Mixer setting for LDAC to LOUT & RDAC to ROUT
 WriteReg(0x2A, 0xB8);

 WriteReg(0x02, 0x00);   //startup FSM and DLL
 DelayMS(500);
 WriteReg(0x2e, 0x1C);
 WriteReg(0x2f, 0x1C);
 WriteReg(0x30, 0x1C);
 WriteReg(0x31, 0x1C i);

 WriteReg(0x04, 0x3C);   //Enable LOUT&ROUT
 WriteReg(0x03, 0x09);   //Power up ADC, Enable LIN&RIN, Power down MICBIAS, set int1lp to low power mode
```

11.2 THE SEQUENCE FOR START UP RECORDING

```
void ES8388_ADC(void)
{
 unsigned int i;
 WriteReg(0x01, 0x58);     //Power down whole chip analog
 WriteReg(0x01, 0x50);     //power up whole chip analog
 WriteReg(0x02, 0xF3);     //stop STM and DLL, power down DAC&ADC vref
 WriteReg(0x02, 0xF0);     //power up DAC&ADC vref
 WriteReg(0x2B, 0xc0);     //set internal ADC and DAC use the same LRCK clock, ADC LRCK as internal LRCK
 WriteReg(0x00, 0x16);     //ADC clock is same as DAC. Use ADC MCLK as internal MCLK

  #if defined(CODEC_I2S_MASTER_MODE)
     WriteReg(0x08, 0x80);   //CODEC IN I2S MASTER MODE
 #elif defined(CODEC_I2S_SLAVE_MODE)
     WriteReg(0x08, 0x00);   //CODEC IN I2S SLAVE MODE
 #endif

 WriteReg(0x04, 0xC0);   //power down DAC, Disable LOUT&ROUT
 WriteReg(0x05, 0x00);   //low power setting
 WriteReg(0x06, 0xC3);

 #if defined(SINGLE_ENDED_MIC)     //Single ended microphone input
     WriteReg(0x0A, 0x00);   //select LIN1&RIN1 as Microphone
     //WriteReg(0x0A, 0x50); // select LIN2&RIN2 as Microphone
#elif defined(DIFF_ENDED_MIC)        //differential microphone input
      WriteReg(0x0A, 0xF0);   //set differential input mode
      WriteReg(0x0B, 0x02); // select LIN1&RIN1 as Microphone
      //WriteReg(0x0B, 0x82); //elect LIN2&RIN2 as Microphone
 #endif

 WriteReg(0x0C, 0x00);   //ADC I2S-24BIT
 WriteReg(0x0D, 0x02);   //ADCLRCK = MCLK/256

 WriteReg(0x10, 0x00);   //ADC VOLUME = 0DB
WriteReg(0x11, 0x00);

WriteReg(0x09, 0x88);     //MIC PGA =24DB
WriteReg(0x12, 0xE2);     //MIC ALC SETTING
WriteReg(0x13, 0xC0);
WriteReg(0x14, 0x12);
WriteReg(0x15, 0x06);
WriteReg(0x16, 0xC3);

WriteReg(0x02, 0x55);   //startup FSM and DLL
WriteReg(0x03, 0x09);   //Power up ADC, Enable LIN&RIN, Power down MICBIAS, set int1lp to low power mode
}
```

11.3 THE SEQUENCE FOR START UP PLAY BACK MODE

```
void ES8388_DAC(void)
{
 unsigned int i;
 WriteReg(0x01, 0x58);      //Power down whole chip analog
 WriteReg(0x01, 0x50);      //power up whole chip analog
 WriteReg(0x02, 0xF3);      //stop STM and DLL, power down DAC&ADC vref
 WriteReg(0x02, 0xF0);      //power up DAC&ADC vref
 WriteReg(0x2B, 0x80);      //set internal ADC and DAC use the same LRCK clock, dac LRCK as internal LRCK
 WriteReg(0x00, 0x36);      //ADC clock is same as DAC. Use DAC MCLK as internal MCLK

  #if defined(CODEC_I2S_MASTER_MODE)
     WriteReg(0x08, 0x80);   //CODEC IN I2S MASTER MODE
 #elif defined(CODEC_I2S_SLAVE_MODE)
     WriteReg(0x08, 0x00);   //CODEC IN I2S SLAVE MODE
 #endif

 WriteReg(0x04, 0x00);   //power up DAC
 WriteReg(0x05, 0x00);   //low power setting
 WriteReg(0x06, 0xC3);

 WriteReg(0x17, 0x00);    //DAC I2S-24BIT

 WriteReg(0x18, 0x02);    //DACLRCK = MCLK/256

 WriteReg(0x1A, 0x00);   //DAC VOLUME = 0DB
 WriteReg(0x1B, 0x00);

 WriteReg(0x27, 0xB8);      //Mixer setting for LDAC to LOUT & RDAC to ROUT
 WriteReg(0x2A, 0xB8);

 WriteReg(0x02, 0xAA);   //startup FSM and DLL
 DelayMS(500);
 WriteReg(0x2e, 0x1C);
 WriteReg(0x2f, 0x1C);
 WriteReg(0x30, 0x1C);
 WriteReg(0x31, 0x1C);

 WriteReg(0x04, 0x3C);   //Enable LOUT&ROUT
 WriteReg(0x03, 0xFF);   //Power DOWN ADC, Disable LIN&RIN, Power down MICBIAS, set int1lp to low power mode
```

11.4 THE SEQUENCE FOR START UP BYPASS MODE

```
void ES8388_BypassMode(void)
{
 unsigned int i;
 WriteReg(0x01, 0x58);      //Power down whole chip analog
 WriteReg(0x01, 0x50);      //power up whole chip analog
 WriteReg(0x02, 0xF3);      //stop STM and DLL, power down DAC&ADC vref

WriteReg(0x02, 0xF0);      //power up DAC&ADC vref
WriteReg(0x2B, 0x80);      //set internal ADC and DAC use the same LRCK clock, dac LRCK as internal LRCK
WriteReg(0x00, 0x36);      //ADC clock is same as DAC. Use DAC MCLK as internal MCLK

#if defined(CODEC_I2S_MASTER_MODE)
     WriteReg(0x08, 0x80);   //CODEC IN I2S MASTER MODE
#elif defined(CODEC_I2S_SLAVE_MODE)
     WriteReg(0x08, 0x00);   //CODEC IN I2S SLAVE MODE
#endif

WriteReg(0x04, 0xC0);   //power up DAC
WriteReg(0x03, 0xFF);   //Power DOWN ADC, Disable LIN&RIN, Power down MICBIAS, set int1lp to low power mode


WriteReg(0x26, 0x00);    // If audio on LIN1&RIN1
```

```
    // WriteReg(0x26, 0x09);    //If audio on LIN2&RIN2
    WriteReg(0x27, 0x50);       //Mixer setting for LIN to LOUT & RIN to ROUT
    WriteReg(0x2A, 0x50);

    WriteReg(0x02, 0x00);    //startup FSM and DLL
    DelayMS(5);
    WriteReg(0x2e, 0X1C);
    WriteReg(0x2f, 0X1C);
    WriteReg(0x30, 0X1C);
    WriteReg(0x31, 0X1C);

    WriteReg(0x02, 0Xf3);    //stop FSM and DLL
    WriteReg(0x04, 0xFC);    //Enable LOUT&ROUT
```

11.5 POWER DOWN SEQUENCE (TO STANDBY MODE)

```
 void ES8388_PowerDown(void)
{
 unsigned int i;
 WriteReg(0x19,0xE6);
 WriteReg(0x04,0xFC);
 WriteReg(0x03,0xFF);
 WriteReg(0x02,0xC0);
 WriteReg(0x2B,0x90);
}
```

11.6 RESUME FROM STANDBY MODE SEQUENCE

```
void ES8388_PowerUp(void)
{
  unsigned int i;
  WriteReg(0x2B,0x80);
  WriteReg(0x02,0x00);
  WriteReg(0x03,0x00);
  WriteReg(0x04,0x3C);
  WriteReg(0x19,0xE2);
}
```