xcore.ai Multichannel Audio Board 1v1 Hardware Manual

Publication Date: 2025/2/27
Document Number: XM-014727-PC v1.1.1

IN THIS DOCUMENT

The xcore.ai Multichannel Audio Board (XK-AUDIO-316-MC-AB) is a complete hardware and reference software platform providing more than 32 input and 32 output channels simultaneously over USB using the USB Audio Class.

The Multichannel Audio Platform hardware is based around the XU316-1024-TQ128-C24 multicore microcontroller; an xcore.ai device with an integrated High Speed USB 2.0 PHY and 16 logical cores delivering up to 2400MIPS of deterministic and responsive processing power.

Exploiting the flexible programmability of the xcore.ai architecture, the Multichannel Audio Board supports USB streaming up to 32 input and 32 output audio channels simultaneously - at up to 192kHz (full details in *Channel counts*). Ideal for consumer and professional USB audio interfaces. The board can also be used for testing general purpose audio DSP activities - mixing, filtering, etc.

The guaranteed Hardware-Response™ times of xCORE technology always ensure lowest latency (round trip as low as 3ms), bit perfect audio streaming to and from the USB host.

Delivered as source code, the reference software provides a fully featured production ready solution, including support for:

- Full-Speed and High-Speed USB operation, Audio Class 2.0 & 1.0, MIDI, SPDIF, ADAT, HID and DFU classes.

1 Features

A block diagram of the xcore.ai Multichannel Audio Board (XK-AUDIO-316-MC-AB) is shown below:

It includes the following features:

- A: xcore.ai (XU316-1024-TQ128-C24) Multicore Microcontroller device

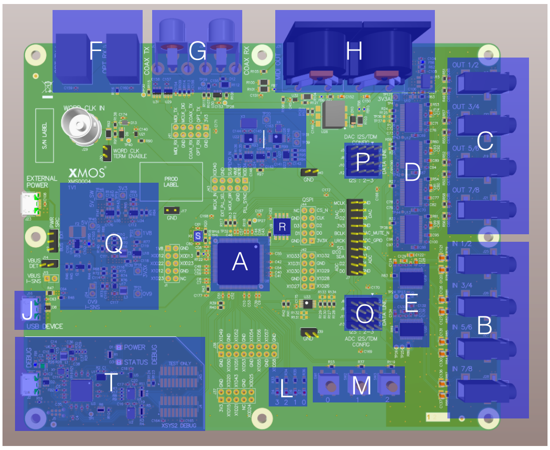


Fig. 1: xcore.ai Multichannel Audio Board block diagram

- B: 8 line level analog inputs (3.5mm stereo jacks)
- C: 8 line level analog outputs (3.5mm stereo jacks)
- D: 384kHz 24 bit audio DACs
- E: 192kHz 24 bit audio ADCs
- F: Optical connections for digital interface (e.g. S/PDIF and ADAT)
- G: Coaxial connections for digital interfaces (e.g. S/PDIF)
- H: MIDI in and out connections
- I: Flexible Audio Master clock generation
- J: USB 2.0 micro-B jacks
- L: 4 general purpose LEDs
- M: 3 general purpose buttons
- O: Flexible I2S/TDM input data routing
- P: Flexible I2S/TDM output data routing
- Q: Integrated power supply
- R: Quad-SPI boot ROM
- S: 24MHz Crystal
- T: Integrated XTAG4 debugger

2 xCORE Multicore Microcontroller Device

The xcore.ai Multichannel Audio board (XK-AUDIO-316-MC-AB) is based on a two-tile xcore.ai device (XU316-1024-TQ128-C24). Each tile is user-programmable, providing eight logical cores with a total of up to 1200 MIPS compute per tile. The device has an integrated high speed USB interface which allows connection to a USB host. The device has a large amount of digital IO which are used for interfacing to the on board components, spare IOs are wired to expansion headers.

For information on xcore.ai follow this link: <https://www.xmos.ai/xcore-ai/>

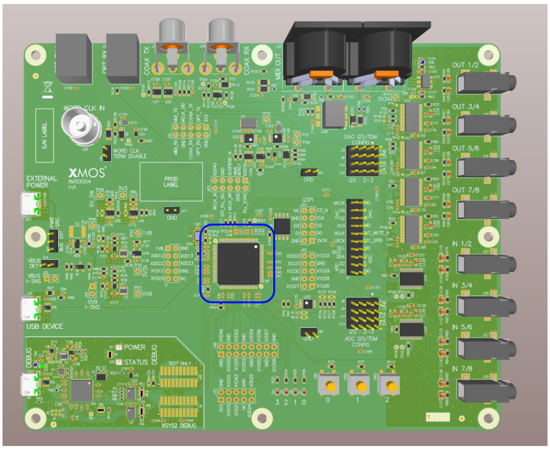


Fig. 2: xcore.ai device

3 Boot

The xcore.ai device boots from an on-board QSPI FLASH memory. The device on the board is 32Mbit in capacity to allow for experimentation but in target applications it is expected a 16Mbit or smaller device would be used to reduce cost.

The FLASH is connected as shown in Table 1:

Table 1: QSPI Flash

Board Net	Pin	Port
QSPI_CS_N	X0D01	P1B0
QSPI_D0	X0D04	P4B0
QSPI_D1	X0D05	P4B1
QSPI_D2	X0D06	P4B2
QSPI_D3	X0D07	P4B3
QSPI_CLK	X0D10	P1C0

The XTC tools include the xflash utility for programming compiled programs into the flash memory. The flash upgrade images can be reprogrammed at run time over USB using the DFU (Device Firmware Upgrade) utilities.

4 Clocking

The device is clocked by an integrated low power crystal oscillator block using an external 24MHz crystal.

5 Debug

The board contains an integrated XTAG4 debugger for loading and debugging programs and programming the on board QSPI flash. This debugger connects to the host via the DEBUG micro b USB connector and is connected to the xcore.ai device through the JTAG interface and also a 2 wire xmos link for high speed debugging with xscope. The debugger is powered separately from the main board and takes all of its power from its USB connector. It only needs to be connected while debugging.

6 Analog audio input

A total of eight single-ended analog input channels are provided via four 3.5mm stereo jacks. The 8 analog inputs are routed to two 4 channel ADCs (PCM1865). The ADCs are configured to output digital audio over I2S or TDM. Configuration of the ADCs is via I2C. The maximum input level is 2.1Vrms.

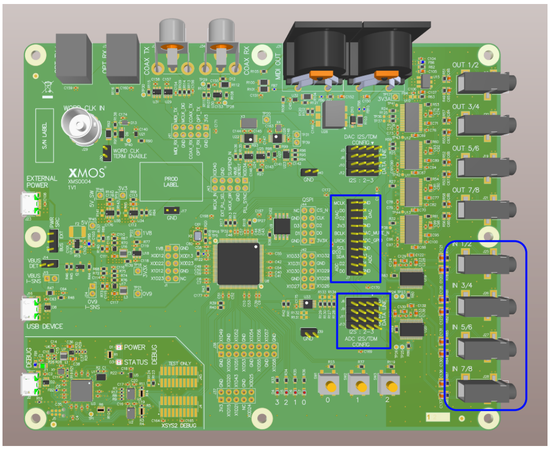


Fig. 3: Analog input stage

The four digital input channels ADC_D0 to ADC_D3 are mapped to the xCORE inputs X_ADC_D0 to X_ADC_D3 through a header array as described in Table 2. Basic jumper operation allows configuring the device to work in I2S or TDM mode. In I2S mode, each wire carries 2 channels of audio so the 8 analog input channels map 1:1 to the four digital inputs. In TDM mode, a single wire can carry up to 8 channels of data. This means the four digital inputs can support up to 32 analog input channels. The jumpers allow the common TDM line from the 8 input channels to connect to any one of the four digital inputs on the xcore.

Table 2: Analog input configuration

Mode	Shorting Jumper Position			
	J6	J9	J11	J13
I2S	2-3	2-3	2-3	2-3
TDM input on X_ADC_D0	1-2, 3-4	NO FIT	1-2	NO FIT
TDM input on X_ADC_D1	1-2	3-4	1-2	NO FIT
TDM input on X_ADC_D2	1-2	NO FIT	1-2, 3-4	NO FIT
TDM input on X_ADC_D3	1-2	NO FIT	1-2	3-4

Pin 1 of each jumper is denoted by a triangle on the silkscreen.

The ADC registers are accessed via the I2C bus - see *I2C bus address map*.

Table 3: Analog input xCORE GPIO

Board Net	xCORE GPIO	Port	Description
X_ADC_D0	X1D24	P1I0	Serial data input 0 (I2S or TDM)
X_ADC_D1	X1D25	P1J0	Serial data input 1 (I2S or TDM)
X_ADC_D2	X1D34	P1K0	Serial data input 2 (I2S or TDM)
X_ADC_D3	X1D35	P1L0	Serial data input 3 (I2S or TDM)
ADC_GPIO	X0D33	P4E3	Multipurpose interrupt input
LRCK	X1D01	P1B0	Serial left/right frame clock
BCLK	X1D10	P1C0	Bit clock for serial data transfer
MCLK_ADC	See note	NA	Buffered global audio master clock

Notes:

- Details of the audio clocking scheme can be found in *Audio clocking*.

7 Analog audio output

A total of eight single-ended analog output channels are provided via four 3.5mm stereo jacks. The 8 analog outputs are generated by four 2 channel DACs (PCM5122). The DACs are configured to accept digital audio over I2S or TDM. Configuration of the DACs is via I2C. The full scale output level is 2.1Vrms.

The four digital input channels ADC_D0 to ADC_D3 are mapped to the xCORE outputs X_DAC_D0 to X_DAC_D3 through a header array as described in Table 4. Basic jumper operation allows configuring the device to work in I2S or TDM mode. In I2S mode, each wire carries 2 channels of audio so the 8 analog output channels map 1:1 to the four digital outputs. In TDM mode, a single wire can carry up to 8 channels of data. This means the four xCORE digital outputs can support up to 32 analog output channels. The jumpers allow the common TDM line to the 8 output channels to connect to any one of the four digital outputs on the xcore.

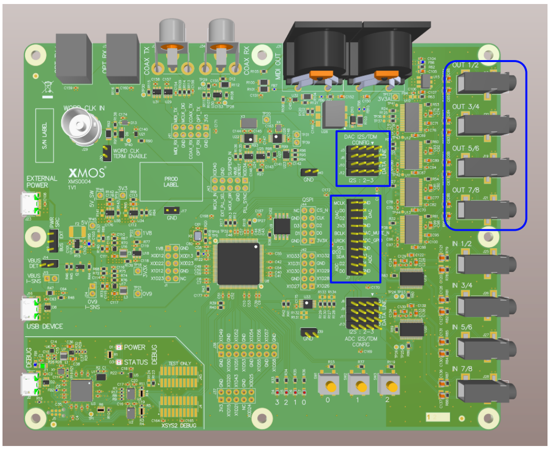


Fig. 4: Analog output stage

Table 4: Analog output configuration

Mode	Shorting Jumper Position			
	J3	J8	J10	J12
I2S	2-3	2-3	2-3	2-3
TDM output from X_DAC_D0	1-2, 3-4	1-2	1-2	1-2
TDM output from X_DAC_D1	1-2	1-2, 3-4	1-2	1-2
TDM output from X_DAC_D2	1-2	1-2	1-2, 3-4	1-2
TDM output from X_DAC_D3	1-2	1-2	1-2	1-2, 3-4

Pin 1 of each jumper is denoted by a triangle on the silk-screen.

The DAC registers are accessed via the I2C bus - see *I2C bus address map*.

Table 5: Analog output xCORE GPIO

Board Net	xCORE GPIO	Port	Description
X_DAC_D0	X1D39	P1P0	Serial data output 0 (I2S or TDM)
X_DAC_D1	X1D38	P1O0	Serial data output 1 (I2S or TDM)
X_DAC_D2	X1D37	P1N0	Serial data output 2 (I2S or TDM)
X_DAC_D3	X1D36	P1M0	Serial data output 3 (I2S or TDM)
DAC_MUTE_N	X0D16	P4D0	Output to mute audio when low
LRCK	X1D01	P1B0	Serial left/right frame clock
BCLK	X1D10	P1C0	Bit clock for serial data transfer
MCLK_DAC	See note	NA	Buffered global audio master clock

Notes:

- Details of the audio clocking scheme can be found in *Audio clocking*.

8 Digital audio output

Optical and coaxial digital audio outputs are provided. Both of these outputs can support transmission of audio data in IEC60958 consumer mode (S/PDIF) format. This format allows transmission of two channels of audio data at up to 192kHz. Note the optical transmitter used on the board can only support up to 96kHz in this mode.

Additionally the optical output can transmit data in the ADAT optical format. This format allows transmission of eight channels of audio data at 48kHz. The S/MUX extension to ADAT allows four channels at 96kHz or two channels at 192kHz.

In order to reduce clock jitter on these outputs, the data outputs from xcore.ai are re-clocked using the global audio master clock to synchronise the data into the audio clock domain. This is achieved using simple external D-type flip-flops.

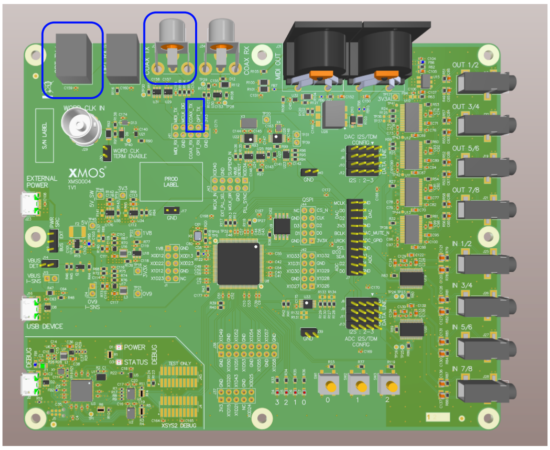


Fig. 5: Digital audio output

The signals are generated from two 1-bit ports on xcore.ai.

Table 6: Digital audio output xCORE GPIO

Board Net	xCORE GPIO	Port	Description
OPT_TX	X1D22	P1G0	IEC60958 (S/PDIF) or ADAT optical output
COAX_TX	X1D00	P1A0	IEC60958 (S/PDIF) output
MCLK_DIG	See notes below		Buffered global audio master clock

Notes:

- Details of the audio clocking scheme can be found in *Audio clocking*.

9 Digital audio input

Optical and coaxial digital audio input is provided. Both of these inputs can support reception of audio data in IEC60958 consumer mode (S/PDIF) format. This format allows transmission of two channels of audio data at up to 192kHz. Note the optical receiver used on the board can only support up to 96kHz in this mode.

Additionally the optical input can receive data in the ADAT optical format. This format allows transmission of eight channels of audio data at 48kHz. The S/MUX extension to ADAT allows four channels at 96kHz or two channels at 192kHz.

The coaxial input receiver circuit uses an unbuffered logic inverter configured as an analog amplifier. This allows the 0.5V pk-pk signal to be amplified up to logic levels required by xcore.ai. The 100k feedback resistor connected between input and output biases the input stage to work in the analog region. This circuit does provide a low cost solution but can use high supply currents (even when no input is present) up to 20mA or more. The addition of the 68 ohm resistor in the supply line to the logic gate helps to reduce this current significantly without affecting circuit operation. It works by reducing the supply voltage to the device to around 2.5V which significantly reduces crossover current. The output voltage will now only swing to 2.5V in the high state but this actually matches quite well with the xcore.ai inputs which have a switching threshold around 1.3V typically. Many other types of receive circuit are possible.

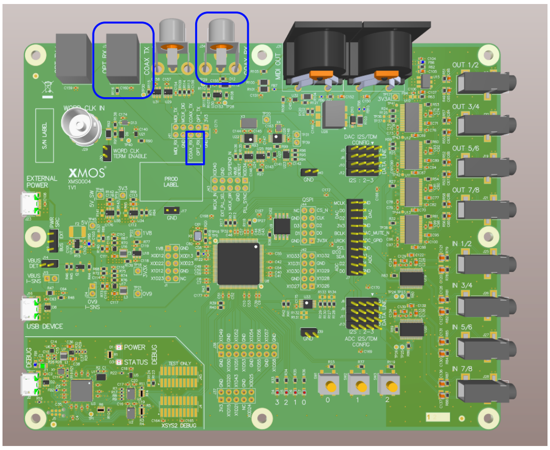


Fig. 6: Digital audio input

The input signals are fed into two 1-bit ports on the xcore.ai device.

Table 7: Digital audio input xCORE GPIO

Board Net	xCORE GPIO	Port	Description
OPT_RX	X0D38	P1O0	IEC60958 (S/PDIF) or ADAT optical input
COAX_RX	X0D37	P1N0	IEC60958 (S/PDIF) input

10 MIDI input and output

MIDI input and output is provided on the board via standard 5-pin DIN connectors compliant to the MIDI specification.

The MIDI signalling is essentially a 31.25kbaud UART over a 5mA current loop. The interface circuits on this board are compliant with the MIDI specification addendum pertaining to use with 3.3V logic signalling “MIDI 1.0 Electrical Specification Update 1.1 [2014]”.

The 3.3V logic signals are connected directly to ports on xcore.ai. A pullup resistor is used to ensure the MIDI output does not drive when the xcore.ai device is not actively driving the output.

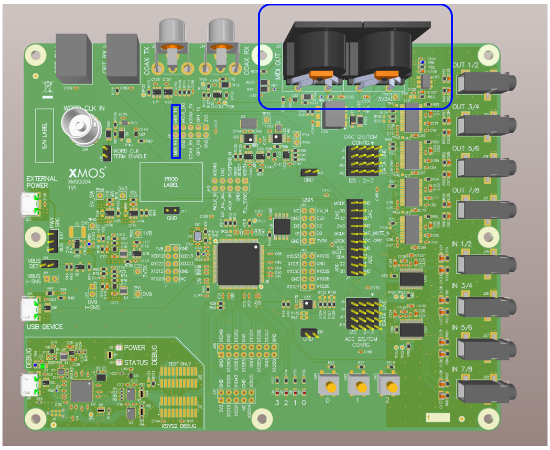


Fig. 7: MIDI input/output

The MIDI input signal is connected to a 1-bit port on xcore.ai, the MIDI output signal is driven by bit 0 of a 4-bit port, the other bits of the port being unused. This is described in Table 8.

Table 8: MIDI xCORE GPIO

Board Net	xCORE GPIO	Port	Description
MIDI_TX	X1D14	P4C0	MIDI output signal
MIDI_RX	X1D13	P1F0	MIDI input signal

11 Audio clocking

A flexible clocking scheme is provided to allow for experimentation with different clocking architectures.

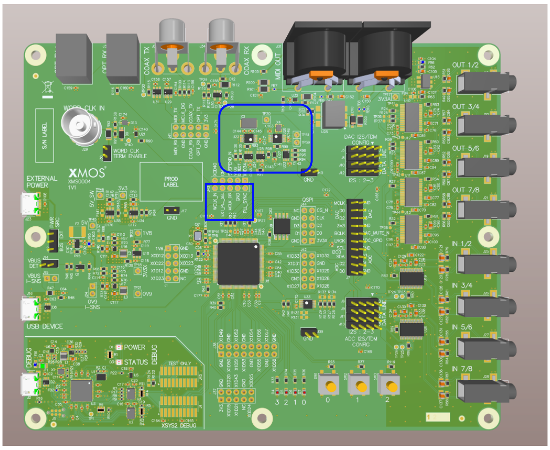


Fig. 8: Clocking circuit

The audio master clock can be generated from one of three possible sources: the xcore.ai secondary (application) PLL, a Cirrus Logic CS2100 Fractional-N clock multiplier, or a Skyworks Si5351A-B-GT CMOS clock generator.

The master clock source is chosen by driving two control signals as shown below:

Control Signal		Master Clock Source
EXT_PLL_SEL	MCLK_DIR	
0	0	Cirrus CS2100
1	0	Skyworks SI5351A-B-GT
X	1	xcore.ai secondary (application) PLL

Each of the sources have potential benefits, some of which are discussed below:

- The Cirrus CS2100 simplifies generating a master clock locked to an external clock (such as S/PDIF in or word clock in).
  - It multiplies up the PLL_SYNC signal which is generated by the xcore.ai device based on the desired external source (so S/PDIF in frame signal or word clock in).
- The Si5351A-B-GT offers very low jitter performance at a relatively lower cost than the CS2100. Locking to an external source is more difficult.
- The xcore.ai application PLL is obviously the lowest cost and significantly lowest power solution, however its jitter performance can not match the Si5351A which may be important in demanding applications. Locking to an external clock is possible but involves more complicated firmware and more MIPS.

In fixed mode (not locking to external source) one example of the application PLL jitter performance on the xcore.ai multichannel audio board is shown below. Divided versions of these clock frequencies will have similar jitter levels.

Desired Frequency		49.152MHz (1024x48kHz)	45.1584MHz (1024x44.1kHz)
Actual Frequency		45.157895MHz	49.151786MHz
Frequency Error		-11.2ppm	-4.4ppm
Baseband Jitter	100Hz - 40kHz	7ps	7ps
	100Hz - 1MHz	67ps	70ps
	100Hz corner	215ps	118ps

Frequency accuracy of 0ppm is possible for most clock frequencies but they are likely to have higher jitter levels. The frequency accuracies in this example (<11ppm) are better than would be expected of typical crystal oscillators so would only be an issue in systems requiring synchronicity from multiple devices on a common 24MHz clock reference which is quite rare.

Clock jitter of 7ps in the 100Hz - 40kHz baseband shows the ability to produce very high quality DAC/ADC performance in the 0-20kHz audio band (jitter artefacts should be below the noise floor for the majority of converters).

Of course a fixed audio master clock is only suitable for certain applications: Asynchronous mode USB audio or standalone analog audio processing. For most other applications (Synchronous or Adaptive mode USB audio, ext S/PDIF, ADAT or Word clock in sync), the local master clock will need to be able to lock to an external clock rate. The Cirrus Logic CS2100 makes this easy as it can generate a low jitter master clock output which is a multiple of a low frequency input reference. The input reference can then come from the clock we need to sync to e.g. word clock in, a signal that toggles every S/PDIF frame or the USB start of frame for Synchronous mode USB.

The xcore.ai application PLL can also work in locking to external clock sources but it requires a software PLL to be implemented which will use some processing power. Work on this solution is ongoing.

The Skyworks Si5351A-B-GT device was included on the board as a fixed very low jitter clock source to use for comparison in audio quality measurements. It has the potential to be used in locking to external sources using a software PLL and this approach is being investigated.

The selected master clock is buffered and distributed to the DAC circuitry, the ADC circuitry and the digital output circuitry.

Audio interface clocks LRCK and BCLK are used in both I2S and TDM modes. These clocks are outputs when the xcore is I2S/TDM master and inputs when xcore is I2S/TDM slave. As outputs these clocks are divided down versions of the audio master clock. As inputs these clocks must be generated by a source synchronous to the global audio master clock.

Table 9: Audio clocking GPIO

Board Net	xCORE GPIO	Port	Description
MCLK_DIR	X0D43	P8D7	Controls audio master clock direction. MCLK is an input to xcore when low and an output when high.
EXT_PLL_SEL	X0D42	P8D6	Selects which external PLL to use. CS2100 when low, Si5351A-B-GT when high.
PLL_SYNC	X0D00	P1A0	Output to CS2100 frequency reference input.
MCLK_XMOS	X1D11	P1D0	Audio master clock. Input or output as specified by MCLK_DIR and internal firmware config.
MCLK_XMOS	X0D11	P1D0	Audio master clock. Input only - for use by tile0 threads e.g. USB thread for clock sync.
LRCK	X1D01	P1B0	Left/Right Frame clock. Input or output.
BCLK	X1D10	P1C0	Serial bit clock. Input or output.

The Cirrus Logic CS2100 and Skyworks Si5351A-B-GT devices are controlled using I2C. Further information on the xcore.ai Multichannel Audio Board I2C bus can be found in *I2C bus address map*.

12 USB Device

As a USB device, the board is connected to the host by the USB micro B connector marked "USB DEVICE".

The board can be configured in firmware as a bus or self powered USB device. In addition to the firmware, two jumpers should be configured on the board to support the relevant mode.

USB Device Type	Jumper Setting	
	J22 "PWR SRC"	J14 "VBUS DET"
Bus Powered	1-2 "BUS"	No Fit
Self Powered	2-3 "EXT"	Fit

The "PWR SRC" jumper selects the power source for the board. When set to bus powered, power is taken from the "USB DEVICE" Vbus pin. When set to self powered, power is taken from the "EXTERNAL POWER" Vbus pin.

The "VBUS DET" jumper enables detection of the presence of Vbus which is required only when the board is configured as a self powered device. This is used to disable the pullup connected to D+ when the host is unpowered, a requirement of the USB specification.

The Vbus detection circuit uses a bipolar transistor to detect the presence of Vbus and output this as a 3.3V logic signal. This circuit is an improvement from a simple resistor divider connected to an IO as it avoids any current paths from Vbus to the power supplies on the board which could cause back powering of these supplies from Vbus (in the case where USB is connected to a host but the device is unpowered). When directly using an IO this path would be present due to the ESD clamp diodes from the IO to the IO power supply. Back powering the board in this way can cause the supplies not to come up cleanly and so some devices with built in power on reset may be affected.

Table 10: USB xCORE GPIO

Board Net	xCORE GPIO	Port	Description
VBUS_DETECT	X0D14	P4C0	VBUS presence detect. High if Vbus is less than ~0.6V

13 General purpose user interface

4 green LEDs and 3 buttons are provided for general purpose user interfacing.

The figure below shows the layout of the user interface subsection:

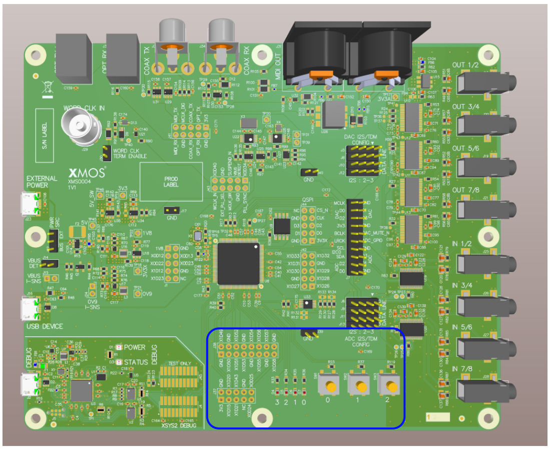


Fig. 9: User interface components

The signal mapping of the user interface components is shown in Table 11

Table 11: User interface GPIO

Board Net	xCORE GPIO	Port	Description
LED_0	X0D28	P4F0	LED Output 0. Set high to light LED.
LED_1	X0D29	P4F1	LED Output 1. Set high to light LED.
LED_2	X0D30	P4F2	LED Output 2. Set high to light LED.
LED_3	X0D31	P4F3	LED Output 3. Set high to light LED.
BUT_0	X0D26	P4E0	Button input 0. High = not pressed.
BUT_1	X0D27	P4E1	Button input 1. High = not pressed.
BUT_2	X0D32	P4E2	Button input 2. High = not pressed.

14 I2C bus address map

Due to a conflict in I2C slave addresses, two I2C buses are used on the board. These buses are multiplexed onto the xcore.ai I2C bus using a PCA9540B 2-channel I2C-bus multiplexer. This allows the I2C bus to be used (0 or 1) to be selected over I2C.

I2C slave address (7 bit)	Device
0x70	NXP PCA9540B 2-channel I2C-bus multiplexer

The I2C multiplexer must be written first to select bus 0 or bus 1 before communication with the I2C devices on the relevant I2C bus can commence.

The I2C multiplexer takes the last byte written as the data for the control register. For an I2C write we typically send a three byte string of slave address, register address, data. In this case we do not use the register address so this can be set to 0. The control register data is set to 0x04 to select I2C bus 0 and set to 0x05 to select I2C bus 1.

So to select bus 0 - WRITE 0x70 0x00 0x04. To select bus 1 - WRITE 0x70 0x00 0x05

A table showing details on I2C devices on the board can be seen here Table 12

Table 12: I2C devices

I2C Bus	I2C slave address (7 bit)	Device
0	0x4C	TI PCM5122 stereo audio DAC 0 (Ch1/2)
	0x4D	TI PCM5122 stereo audio DAC 1 (Ch3/4)
	0x4E	TI PCM5122 stereo audio DAC 2 (Ch5/6)
	0x4F	TI PCM5122 stereo audio DAC 3 (Ch7/8)
	0x4A	TI PCM1865 4-Channel audio ADC 0 (Ch1/2/3/4)
	0x4B	TI PCM1865 4-Channel audio ADC 1 (Ch5/6/7/8)
1	0x4E	Cirrus Logic CS2100 Fractional-N clock multiplier
	0x60	Skyworks Si5351A-B-GT CMOS clock generator

15 Power supplies

The board is powered either from the “USB DEVICE” connector or from the “EXTERNAL POWER” connector as described in the USB section. This allows for testing as a bus or self powered USB device.

The USB specification has quite stringent requirements as to how the device uses power from Vbus. Some of these are listed below:

- Device must present less than 10uF of decoupling capacitance on Vbus.
- Device must use less than 2.5mA from Vbus in USB suspend.
- Device must use less than 100mA from Vbus until configured as a high power device.

For self powered devices these are trivially met as the board can use as much power as required from the external power supply. For bus powered devices, the xcore.ai multichannel audio board is designed to meet these requirements where possible which makes for a slightly more complicated power supply scheme.

The essence of the design is that only supplies required by the xmos device are always on and the other power supplies on the board are switched off until they are required/allowed. An additional complication is switching supplies on can cause significant inrush current if not controlled so current limited load switches are used for the purposes of switching supplies on and off rather than regulator enable pins. It should be noted switching the power supplies on and off is not required in all designs, for simple designs the power may be reduced sufficiently by setting all active devices into suspend or power down states.

An OnSemi NCP360 overvoltage protection device is used for two potential benefits: protection against overvoltage transients (e.g. inductive spike on connection of long cable) and also inrush current limiting.

3V3X, 1V8 and 0V9 are the always on supplies for xcore.ai and essential support components. These use dc-dc supplies from the 5V supply apart from 1V8 which uses a low drop out linear regulator from the 3V3X supply. This is because 1V8 uses much less current.

The supplies are sequenced such that the 3V3X and 1V8 supplies will be present before the 0V9 supply is turned on. This meets the requirement of the xcore.ai device that the VDDIOB18 supply must not be turned on last. For more details of supply requirements see the XU316-1024-TQ128 datasheet.

The main board 3V3 and 5V supplies are switched such they can be turned off in low power modes when bus powered. The switched 5V supply is used to generate a dedicated analog 3V3 supply which is used by the DACs and ADCs as their analog supply. This supply is separated and generated by a linear regulator to ensure it is low ripple/noise.

The requirement to use less than 2.5mA when in the USB suspend state is a difficult one to meet. To meet this requirement care must been taken to minimise any unwanted current draw, steps taken on this board are:

- All supplies not required for USB operation are turned off when in suspend.
- The xcore.ai VDD supply voltage is reduced to around 0.85V from the nominal 0.9V.
- All power management components chosen for low quiescent current.
- DC-DC supplies chosen/configured to support high efficiency at low output current (PFM mode).

Table 13: Power Supply xCORE GPIO

Board Net	xCORE GPIO	Port	Description
X0D22	X0D22	P1G0	VDD core supply margin. Drive high for 0.85V, high-z for 0.9V.
SUSPEND_N	X0D41	P8D5	Enables the 5V_SW and 3V3 supplies when high.

16 Operating requirements

This development board is, like most exposed electronic equipment, sensitive to Electrostatic Discharge (ESD) events. Users should operate the xcore.ai multichannel audio board with appropriate ESD precautions in place.

17 Dimensions

The xcore.ai multichannel audio board dimensions are 160x130mm. The mounting holes are 3.2mm in diameter for use with an M3 screw.

18 xcore.ai multichannel audio board port map

The tables below provide a full description of the port-pin-net mappings described throughout this document for the xcore.ai multichannel audio board.

Pin	Port			IO	Board Net	Description
X0D00	P1A0			3.3V	PLL_SYNC	CS2100 Frequency Ref
X0D01	P1B0			3.3V	QSPI_CS_N	Quad SPI Boot
X0D04		P4B0	P8A2	3.3V	QSPI_D0	Quad SPI Boot
X0D05		P4B1	P8A3	3.3V	QSPI_D1	Quad SPI Boot
X0D06		P4B2	P8A4	3.3V	QSPI_D2	Quad SPI Boot
X0D07		P4B3	P8A5	3.3V	QSPI_D3	Quad SPI Boot
X0D10	P1C0			3.3V	QSPI_CLK	Quad SPI Boot
X0D11	P1D0			3.3V	MCLK_XMOS	Audio master clock input
X0D12	P1E0			1.8V	X0D12	GPIO (Used by USB)
X0D13	P1F0			1.8V	X0D13	GPIO (Used by USB)
X0D14		P4C0	P8B0	3.3V	VBUS DETECT	Detect VBUS present
X0D16		P4D0	P8B2	3.3V	DAC_MUTE_N	Hardware DAC Mute
X0D22	P1G0			1.8V	X0D22	Output (1.8V, margin core voltage)
X0D23	P1H0			1.8V	X0D23	GPIO (Used by USB)
X0D24	P1I0			3.3V	X0D24	GPIO (Used by USB)
X0D25	P1J0			3.3V	X0D25	GPIO (Used by USB)
X0D26		P4E0	P8C0	3.3V	BUT_0	Button input
X0D27		P4E1	P8C1	3.3V	BUT_1	Button input
X0D28		P4F0	P8C2	3.3V	LED_0	LED Output
X0D29		P4F1	P8C3	3.3V	LED_1	LED Output
X0D30		P4F2	P8C4	3.3V	LED_2	LED Output
X0D31		P4F3	P8C5	3.3V	LED_3	LED Output
X0D32		P4E2	P8C6	3.3V	BUT_2	Button input
X0D33		P4E3	P8C7	3.3V	ADC_GPIO	ADC interrupt input
X0D34	P1K0			3.3V	X0D34	GPIO (Used by USB)
X0D35	P1L0			3.3V	I2C_SCL	I2C Serial Clock
X0D36	P1M0		P8D0	3.3V	I2C_SDA	I2C Serial Data
X0D37	P1N0		P8D1	3.3V	COAX_RX	S/PDIF input
X0D38	P1O0		P8D2	3.3V	OPT_RX	S/PDIF/ADAT input
X0D39	P1P0		P8D3	3.3V	WCLK_IN	Word Clock input
X0D40			P8D4	3.3V	X0D40	GPIO
X0D41			P8D5	3.3V	SUSPEND_N	Low power mode output
X0D42			P8D6	3.3V	EXT_PLL_SEL	PLL Select output
X0D43			P8D7	3.3V	MCLK_DIR	Master clock direction

Pin	Port			IO	Board Net	Description
X1D00	P1A0			3.3V	COAX_TX	S/PDIF output
X1D01	P1B0			3.3V	LRCK	Serial audio interface LRCK
X1D09		P4A3	P8A7	3.3V	X1D09	GPIO
X1D10	P1C0			3.3V	BCLK	Serial audio interface BCLK
X1D11	P1D0			3.3V	MCLK_XMOS	Audio master clock output/input
X1D12	P1E0			1.8V	X1D12	GPIO
X1D13	P1F0			3.3V	MIDI_RX	MIDI input
X1D14		P4C0	P8B0	3.3V	MIDI_TX	MIDI output
X1D15		P4C1	P8B1	3.3V	X1D15	GPIO
X1D16		P4D0	P8B2	3.3V	XL_DN1	XSCOPE DEBUG
X1D17		P4D1	P8B3	3.3V	XL_DN0	XSCOPE DEBUG
X1D18		P4D2	P8B4	3.3V	XL_UP0	XSCOPE DEBUG
X1D19		P4D3	P8B5	3.3V	XL_UP1	XSCOPE DEBUG
X1D20		P4C2	P8B6	3.3V	X1D20	GPIO
X1D21		P4C3	P8B7	3.3V	X1D21	GPIO
X1D22	P1G0			3.3V	OPT_TX	S/PDIF/ADAT output
X1D23	P1H0			1.8V	X1D23	GPIO
X1D24	P1I0			3.3V	X_ADC_D0	Serial audio interface data input 0
X1D25	P1J0			3.3V	X_ADC_D1	Serial audio interface data input 1
X1D26		P4E0	P8C0	3.3V	X1D26	GPIO
X1D27		P4E1	P8C1	3.3V	X1D27	GPIO
X1D28		P4F0	P8C2	3.3V	X1D28	GPIO
X1D29		P4F1	P8C3	3.3V	X1D29	GPIO
X1D30		P4F2	P8C4	3.3V	X1D30	GPIO
X1D31		P4F3	P8C5	3.3V	X1D31	GPIO
X1D32		P4E2	P8C6	3.3V	X1D32	GPIO
X1D33		P4E3	P8C7	3.3V	X1D33	GPIO
X1D34	P1K0			3.3V	X_ADC_D2	Serial audio interface data input 2
X1D35	P1L0			3.3V	X_ADC_D3	Serial audio interface data input 3
X1D36	P1M0		P8D0	3.3V	X_DAC_D3	Serial audio interface data output 3
X1D37	P1N0		P8D1	3.3V	X_DAC_D2	Serial audio interface data output 2
X1D38	P1O0		P8D2	3.3V	X_DAC_D1	Serial audio interface data output 1
X1D39	P1P0		P8D3	3.3V	X_DAC_D0	Serial audio interface data output 0
X1D43			P8D7	3.3V	X1D43	GPIO

19 Channel counts

The number of audio channels supported depends on many factors that are explained below:

- The board has four stereo codecs, with four stereo line inputs and four stereo line outputs, and hence the board supports up to eight line-in-channels and eight line-out-channels natively.

- The DACs on this board can support up to 384 kHz, the ADCs can support up to 192 kHz. Higher sampling rates, require more bandwidth per channel, and the total bandwidth of the USB stack is limited, as discussed below.
- The USB Audio stack is, at present, limited to 8,192,000 bytes per second per endpoint. At 32-bit sample depth that equates to 2,048,000 samples per second per endpoint. This is 42 channels at 48 kHz, or 10 channels at 192 kHz, or 5 channels at 384 kHz. Alleviating this limit will require custom code that, for example, uses MDATA packets or multiple endpoints.
- The board has jumpers (J3, J6, J8-13), that enable you to intercept the I2S lines, and in software they can be changed to TDM connections, enabling you to add external DACs and ADCs to the board in order to increase channel-count to 32 channels in and 32 channels out. This will need external hardware.

20 xcore.ai Multichannel Audio Board schematics

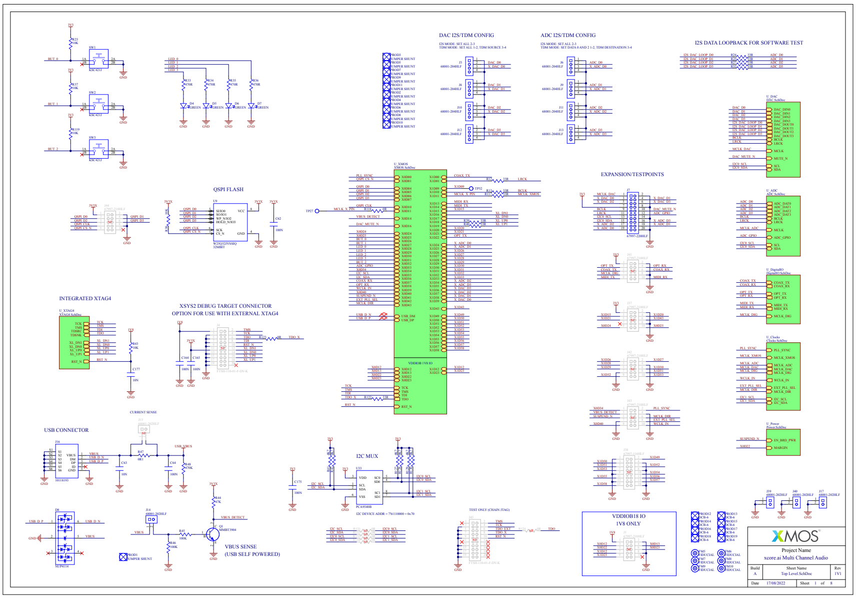


Fig. 10: xcore.ai Multichannel Audio Board schematic (1 of 8)

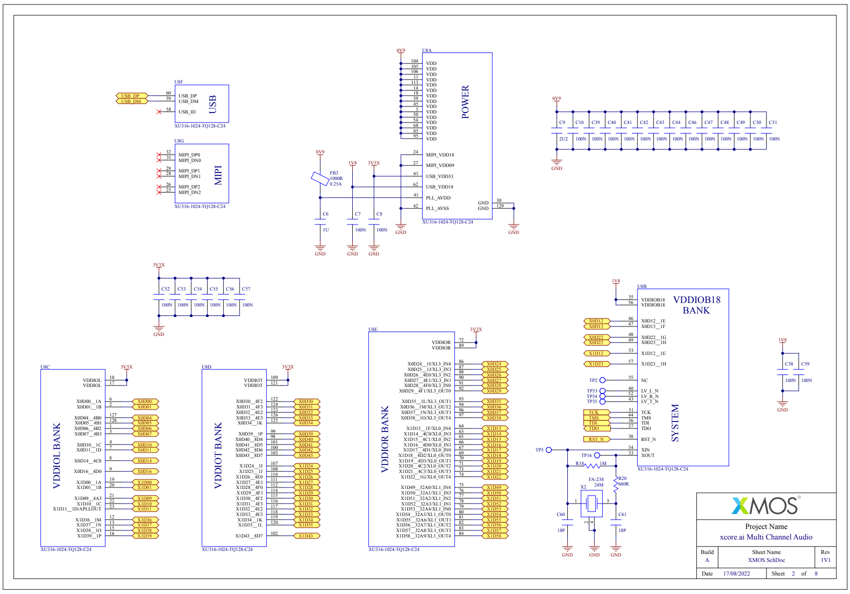
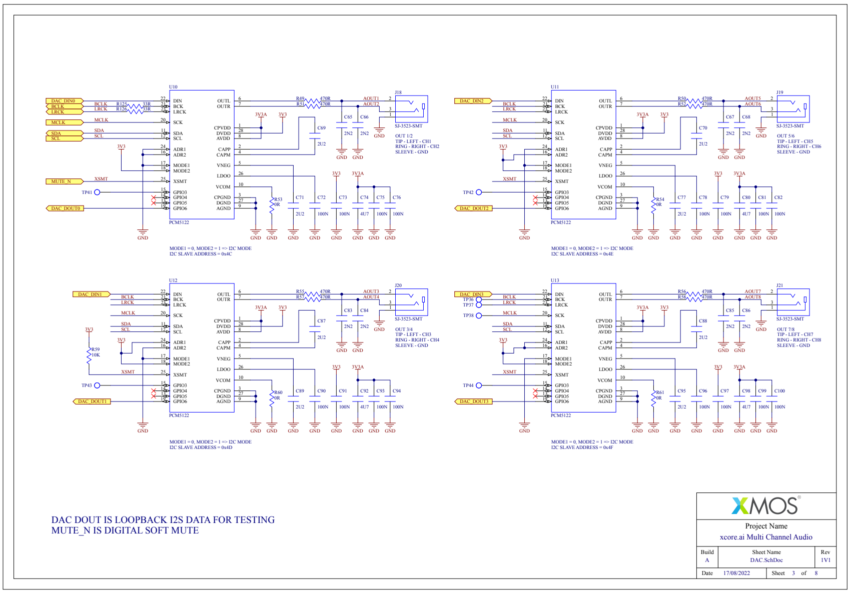


Fig. 11: xcore.ai Multichannel Audio Board schematic (2 of 8)



DAC DOUT IS LOOPBACK I2S DATA FOR TESTING
MUTE_N IS DIGITAL SOFT MUTE

Fig. 12: xcore.ai Multichannel Audio Board schematic (3 of 8)

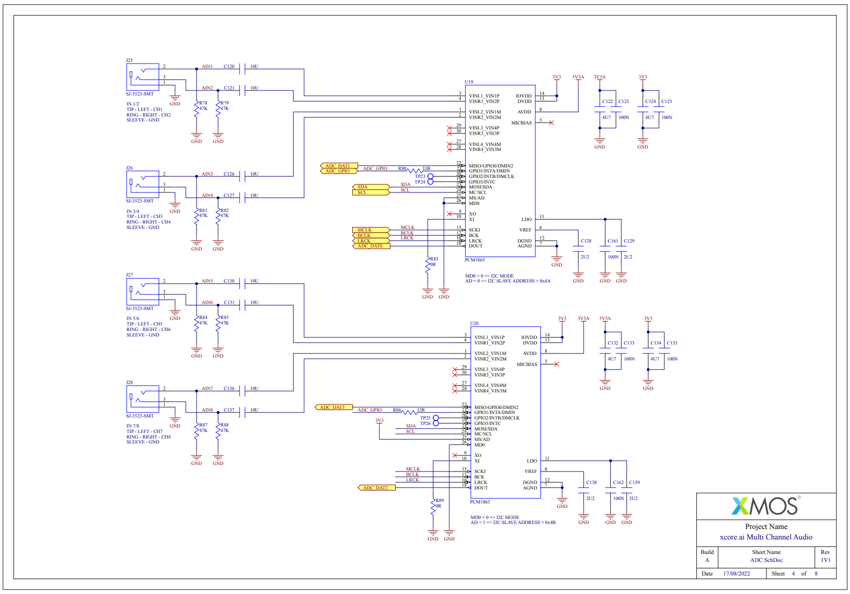


Fig. 13: xcore.ai Multichannel Audio Board schematic (4 of 8)

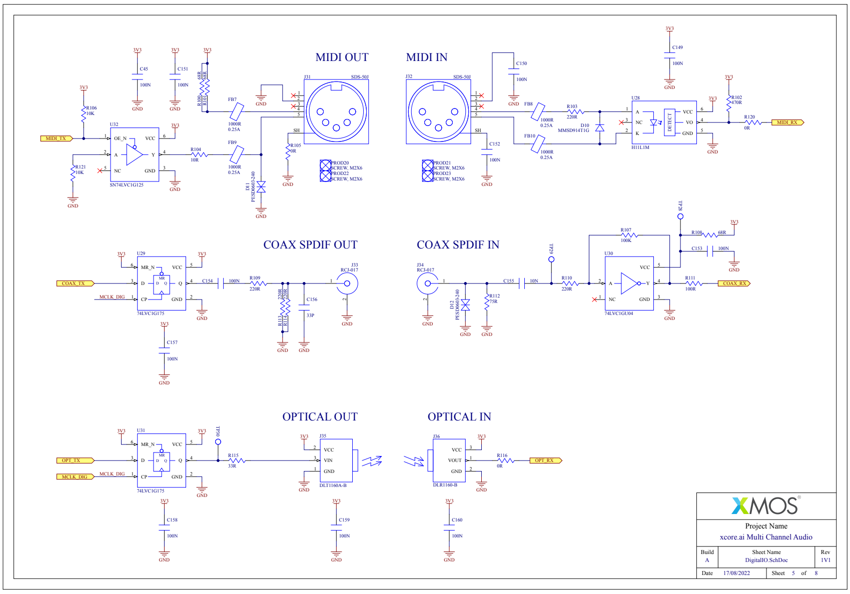
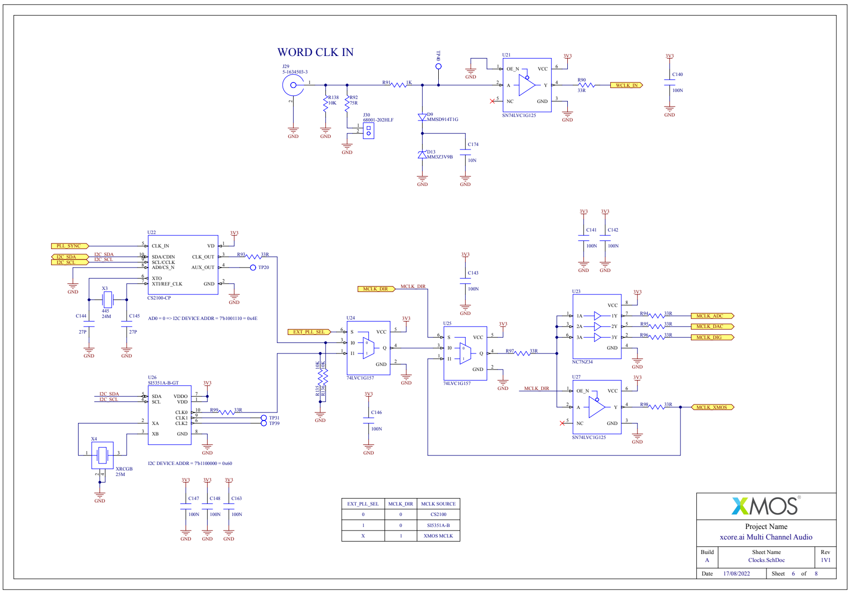


Fig. 14: xcore.ai Multichannel Audio Board schematic (5 of 8)



EXT_PLL_SEL	MCLK_DIR	MCLK SOURCE
0	0	CS2100
1	0	SI5351A-B
X	1	XMOS MCLK

Fig. 15: xcore.ai Multichannel Audio Board schematic (6 of 8)

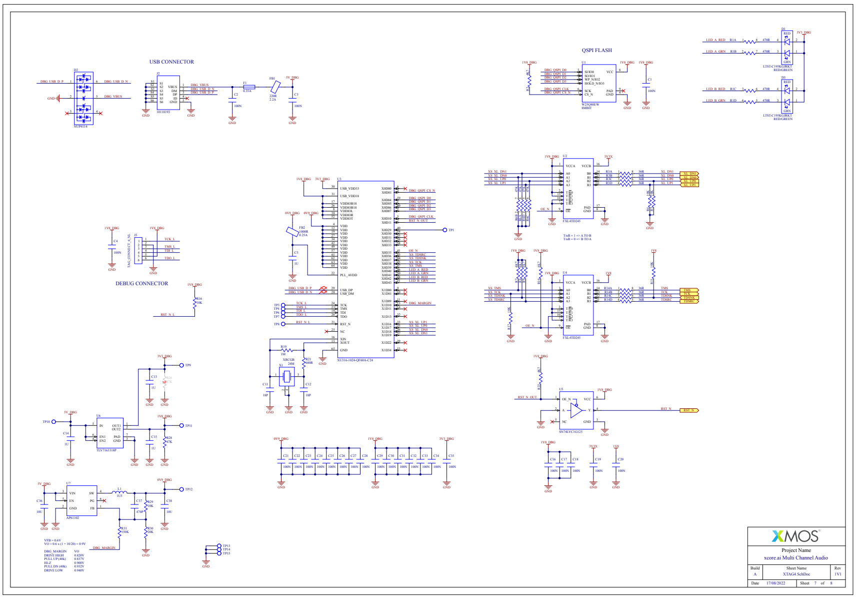


Fig. 16: xcore.ai Multichannel Audio Board schematic (7 of 8)

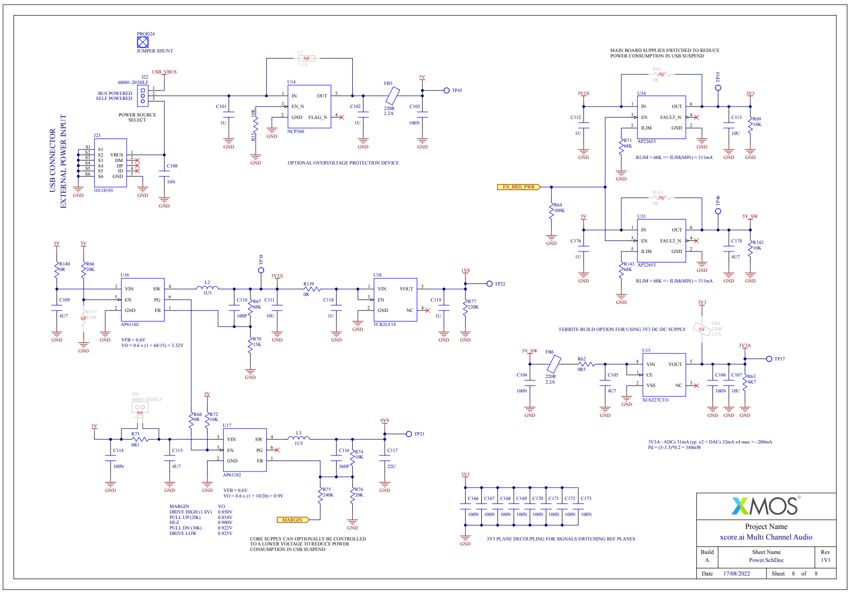
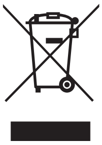

JUMPER SHUNT
BUS POWERED
SELF POWERED
POWER SOURCE SELECT
USB CONNECTOR
EXTERNAL POWER INPUT
OPTIONAL OVERVOLTAGE PROTECTION DEVICE
MAIN BOARD SUPPLIES SWITCHED TO REDUCE POWER CONSUMPTION IN USB SUSPEND
RLIM = 68K => ILIM(MIN) = 311mA
EN_BRD_PWR
VFB = 0.6V
VO = 0.6 x (1 + 68/15) = 3.32V
FERRITE BUILD OPTION FOR USING 3V3 DC-DC SUPPLY
3V3A - ADCs 31mA typ x2 + DACs 32mA x4 max = ~200mA
Pd = (5-3.3)*0.2 = 340mW
VFB = 0.6V
VO = 0.6 x (1 + 10/20) = 0.9V
MARGIN
CORE SUPPLY CAN OPTIONALLY BE CONTROLLED TO A LOWER VOLTAGE TO REDUCE POWER CONSUMPTION IN USB SUSPEND
3V3 PLANE DECOUPLING FOR SIGNALS SWITCHING REF PLANES
XMOS
Project Name
xcore.ai Multi Channel Audio
Build
A
Sheet Name
Power.SchDoc
Rev
1V1
Date 17/08/2022
Sheet 8 of 8

Fig. 17: xcore.ai Multichannel Audio Board schematic (8 of 8)

21 RoHS and REACH

The xcore.ai multichannel audio board complies with appropriate RoHS2 and REACH regulations and is a Pb-free product.

The xcore.ai multichannel audio board is subject to the European Union WEEE directive and should not be disposed of in household waste. Alternative requirements may apply outside of the EU.



22 Version history

Date	Description
2025/2/24	Clarified channel count
2022/9/15	First board release, 1V1 hardware

XMOS Ltd. is the owner or licensee of this design, code, or Information (collectively, the ”Information”) and is providing it to you ”AS IS” with no warranty of any kind, express or implied and shall have no liability in relation to its use. XMOS Ltd makes no representation that the Information, or any particular implementation thereof, is or will be free from any claims of infringement and again, shall have no liability in relation to any such claims.

XMOS, XCORE, VocalFusion and the XMOS logo are registered trademarks of XMOS Ltd. in the United Kingdom and other countries and may not be used without written permission. Company and product names mentioned in this document are the trademarks or registered trademarks of their respective owners.