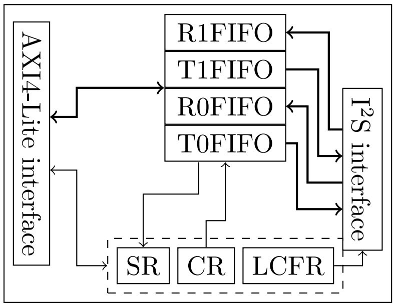
CBI Semi

CBI980AL

AXI4-Lite Audio Interface

Overview

AMBA AXI4-Lite[1]-controlled digital stereo audio input/output device with I^2S[2] master interface



Features

- 32-bit AXI4-Lite control and data interface with 8 registers
- 2 two-way channels with built-in data FIFO for stereo audio RX/TX
- FIFO overrun/underrun detection for I^2S side
- Interrupt capability for all FIFO status flags

Register map

Word Offset	**Byte Offset**	**Register**	**Long Name**	**Access**[a]
0	0000	CVR	Chip ID/Version Register	RO
1	0004	SR	Status Register	RO
2	0008	CR	Control Register	RW
3	000C	LCFR	Line Coding Format Register	RW
4	0010	DOUT1R	Data Output Channel #1 Register	WO
5	0014	DOUT0R	Data Output Channel #0 Register	WO
6	0018	DIN1R	Data Input Channel #1 Register	RO
7	001C	DIN0R	Data Input Channel #0 Register	RO

[a] RO=Read Only, RW=Read/Write, WO=Write Only

Violating access type (i.e. trying to write a RO register), or trying to write < 32 bits at a time will result in a SLVERR. Writing an out-of-range or otherwise invalid value to a register will return OKAY, but the invalid portion of the write will not take effect (i.e. writing unallocated/constant bits in a register will be discarded, but the rest of the register will be updated).

CVR

The CVR holds the chip's identifier, which, for the part covered by this document, is always `0xcb199800`

SR

SR holds the FIFO's status flags, as well as a flag indicating readiness after reset.

Bit	31	30	29	28	27	26	25	24
Value	Init	0						

Bit	23	22	21	20	19	18	17	16
Value	0				RXOVF$_1$	TXUNF$_1$	RXOVF$_0$	TXUNF$_0$

Bit	15	14	13	12	11	10	9	8
Value	RXNE$_1$	RXF$_1$	TXNF$_1$	TXE$_1$	RXNE$_0$	RXF$_0$	TXNF$_0$	TXE$_0$

Bit	7	6	5	4	3	2	1	0
Value	0							

- Init: 0 after reset, 1 when the I^{2}S interface is ready to send data
- RXOVF$_n$: n-th channel RX FIFO overflowed
- TXOVF$_n$: n-th channel TX FIFO underflowed
- RXF$_n$: n-th channel RX FIFO full

- TXNF$_n$: n-th channel TX FIFO not full
- TXE$_n$: n-th channel TX FIFO empty
- RXNE$_n$: n-th channel RX FIFO not empty

CR

CR contains interrupt enable, transmit and receive enable bits, as well as the interrupt flag and a soft reset bit.

Bit	31	30	29	28	27	26	25	24
Value	0							
Bit	23	22	21	20	19	18	17	16
Value	0				IE$_H$			
Bit	15	14	13	12	11	10	9	8
Value	IE$_L$							
Bit	7	6	5	4	3	2	1	0
Value	0		RXEN$_1$	TXEN$_1$	RXEN$_0$	TXEN$_0$	IF/INTc	RST

- IE$_H$, IE$_L$: Interrupt Enable bits, the layout follows the corresponding bits in SR
- RXEN$_n$: n-th channel RX enable
- TXEN$_n$: n-th channel TX enable
- IF/INTc: Interrupt Flag, writing a 1 here clears previously latched RXOVF$_n$ and TXUNF$_n$ bits
- RST: Writing a 1 here performs a soft reset of the device; in this case, other bits will not take effect

Note: INTc will not necessarily clear the interrupt condition; namely, it will not clear RXF$_n$, TXNF$_n$, TXE$_n$, RXNE$_n$ (these stay active so long as the condition exists), nor will it clear RXOVF$_n$ and TXUNF$_n$ on the same clock cycle an over- or underflow occurs.

All flags in CR have a value of 0 after reset.

LCFR

This register ~~does nothing~~ sets the line coding format of the I^{2}S interface.

Bit	31	30	29	28	27	26	25	24
Value	0					MCLK_{rate}		

Bit	23	22	21	20	19	18	17	16
Value	0					SCLK_{rate}		

Bit	15	14	13	12	11	10	9	8
Value	0					#octets		

Bit	7	6	5	4	3	2	1	0
Value	0						R_{just}	LSB_{first}

- MCLK_{rate}: Division factor for MCLK, see MCLK
- #octets: Number of octets per sample to be transmitted, minus one. A value of 0 means 8 bits/sample, 1 means 16 bit, 2 means 24 bit and 3 means 32 bit
- R_{just}: In the case of the 24-bit format, specify alignment within the 32-bit-slot frame. 0 means left-justified (valid data towards the MSB), 1 means right-justified (valid data towards the LSB)
- LSB_{first}: Whether to transmit the Least Significant Bit (LSB) first or last. 0 means MSB first, LSB last, 1 means LSB first, MSB last

Note: According to the I^2S specification[2], data should be MSB first, and – if justification is needed, – left-justified, that is, LSB_{first} and R_{just} should both be 0 (their reset value). These options are there mostly for connecting to non-compliant devices.

The reset values for the fields in LCFR are:

- MCLK_{rate}: 0, meaning a clock division of 2
- #octets: 2, meaning 24-bit mode
- R_{just}: 0, meaning left-justified
- LSB_{first}: 0, meaning MSB transmitted first, LSB last (standard I^2S operation)

Data registers

These 32-bit registers act as the datapath for the audio stream. Reading/writing these registers dequeues/enqueues words from/to the respective FIFOs. In cases where the data to be ent is less than 32 bits, the valid data will occupy the top bits of the register (that is, the input data is expected to be left-justified).

DOUT1R

TX Data Out for channel #1.

DOUT0R

TX Data Out for channel #0.

DIN1R

RX Data In for channel #1.

DIN0R

RX Data In for channel #0.

Clocks

CBI980 is an I^{2}S controller, as per chapter 3 of the specification[2]. The document mandates the presence of two 50% duty cycle clock signals, both generated by the controller:

- SCLK (Serial Clock): it times the data transmission
- LRCLK ($\overline{\text{Left}}$/Right Clock)[1]: it specifies the sample rate (f_S)

On top of that, CBI980 also generates the following optional clock:

- MCLK (Master Clock): it is used by the external I^{2}S codec as a reference

All of the clocks specified here are synthesized from the AXI4-Lite clock (ACLK), with a power-of-two division.

SCLK

SCLK is used for synchronizing the data I/O: new bits are to be sent by the transmitter at its falling edge, and sampled at the receiver on its rising edge.

Internally, SCLK is obtained from ACLK by a frequency division factor of $2^{SCLK_{rate}+1}$, with $SCLK_{rate}$ being specified in LCFR.

LRCLK

LRCLK, on top of specifying the sample rate for the codec, also acts as a channel selector: on the low half-cycle, data for channel #0 is shifted out/in, on the high half-cycle, channel #1's transaction is performed.

LRCLK is synthesized so that it has a frequency of $f_{LRCLK} \geq 8 \cdot \#octets \cdot f_{SCLK}$, while maintaining that $f_{LRCLK} = f_{ACLK} \cdot 2^{-N}$ for some N (the smallest such N that satisfies the first requirement).

MCLK

MCLK is needed by some codecs for a frequency reference base. Its typical frequency is equal to $256 \cdot f_S$. Consult the datasheet of your codec to find the precise value/expected frequency range for your device.

Internally, MCLK is obtained from ACLK by a frequency division factor of $2^{MCLK_{rate}+1}$, with $MCLK_{rate}$ being specified in LCFR.

[1] Called WS (Word Select) by the specification, also sometimes called FS (Frame Sync)

References

[1] ARM: AMBA AXI4-Lite Interface Specification
https://developer.arm.com/documentation/ihi0022/e/AMBA-AXI4-Lite-Interface-Specification

[2] NXP Semiconductors: I^2S bus specification, Rev. 3.0 (2022-02-17)
https://www.nxp.com/docs/en/user-manual/UM11732.pdf