

Die I²S Schnittstelle

1 Einleitung

Die I²S-Schnittstelle wurde 1986 von Philips veröffentlicht und entstand in Zusammenhang mit der Entwicklung von digitalen Audio-Geräten, die zu diesem Zeitpunkt auf dem Markt eingeführt wurden. Der optische Audio-Ausgang am CD-Spieler ist eine Abwandlung der I²S-Schnittstelle. Innerhalb von Geräten findet I²S-Anwendung als Schnittstelle zwischen Mikroprozessoren oder DSPs- und DA/AD-Wandlern. Im DTP2 Projekt wird die I²S-Schnittstelle zwischen dem FPGA und dem Audio-Codec eingesetzt. Abbildung 1 zeigt ein typisches Zeitverlaufdiagramm der I²S-Signale.

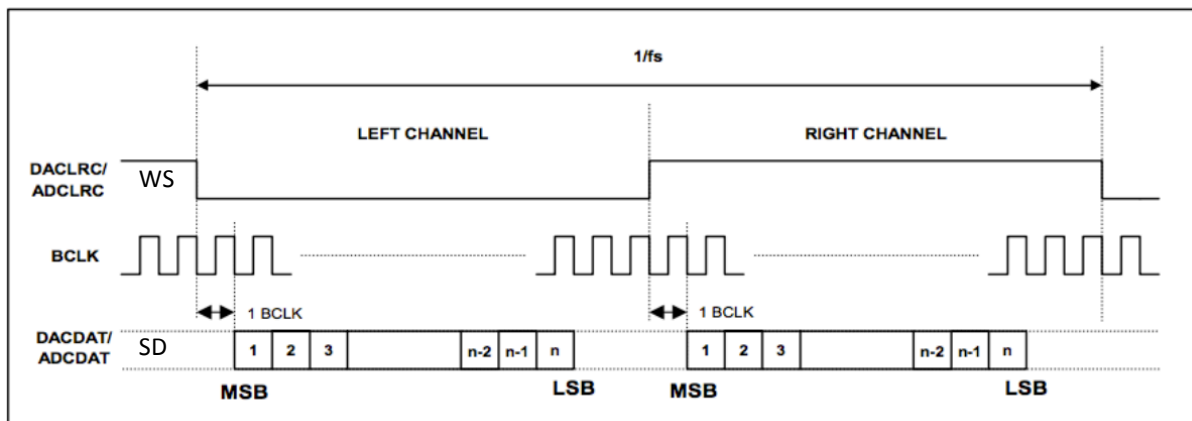


Abbildung 1: Zeitverlaufdiagramm der I²S-Signale

2 Schnittstellenbeschreibung

Die I²S-Schnittstelle beruht auf den Signalen, die in Abbildung 1 ersichtlich sind und im Folgenden erläuterten werden.

2.1 Der I²S serielle Takt (BCLK)

Der Bit Clock (BCLK) ist der Haupttakt der I²S-Schnittstelle. Diese Schnittstelle arbeitet seriell und synchron zu BCLK, wobei die Daten und Steuersignale jeweils auf die Taktflanken von BCLK wechseln. Typischerweise ändern sich die Daten nach jeder fallenden Flanke von BCLK. Jedoch hängt der Zeitverlauf der Daten auch vom Word Select (WS) Signal ab, das weiter unten erklärt wird. Das FPGA stellt dem DA-Wandler im Codec jeweils nach der fallenden Flanke von BCLK ein neues Daten-Bit zur Verfügung. In der anderen Richtung stellt der AD-Wandler im Codec dem FPGA zur gleichen Zeit ein neues Datenbit bereit.

BCLK kann in Bezug zum Codec ein Eingangssignal oder ein Ausgangssignal sein. Wenn BCLK ein Eingangssignal zum Codec ist, spricht man beim Codec von Slave-Mode. Im DTP2 Projekt wird der Codec im Slave-Mode betrieben.

2.2 Die seriellen Datenleitungen (DACDAT/ADCDAT)

Serielle Daten (SD) werden auf zwei Leitungen übertragen (DACDAT/ADCDAT), da sowohl Daten zum Codec gesendet als auch von diesem empfangen werden (digital-zu-analog und analog-zu-digital).

Der Codec unterstützt Stereo-Audio-Ein- und Ausgabe. Deshalb werden abwechselungsweise ein linker und ein rechter Kanal über die seriellen Datenleitungen übertragen.

2.3 Das Word-Select (DACLRC / ADCLRC) Signal

Das gleiche Word Select Signal (WS), auch Left-Right-Clock genannt, wird dem Codec auf zwei verschiedenen Eingängen zur Verfügung gestellt, einmal für den DA-Teil (DACLRC) und einmal für den AD-Teil des Codecs (ADCLRC). Das Signal enthält folgende Informationen:

- Die fallende Flanke von WS signalisiert den Beginn der Übertragung des linken Kanals.
- Während WS low ist, wird der linke Kanal übertragen. Ist WS high, wird der rechte Kanal übertragen.
- Das MSB des linken Kanals wird erst nach einer BCLK-Periode nach der fallenden Flanke von BCLK ausgegeben (siehe Abbildung 1). Nachdem die Daten des linken Kanals komplett übertragen wurde, werden auf den SD-Leitungen bis zum Beginn des nächsten Audio Samples Nullen übertragen. Somit ist die Schnittstelle nicht nur auf eine Bitbreite beschränkt, sondern kann auch eine unterschiedliche Anzahl von Datenbits übertragen. Im DTP2-Projekt werden 16 Datenbits übertragen.
- Die steigende Flanke von WS signalisiert den Beginn der Übertragung des rechten Kanals. Auch hier wird das MSB wiederum eine BCLK Periode nach der fallenden Flanke von BCLK ausgegeben. Nachdem das Daten-Wort des rechten Kanals komplett übertragen wurde, werden auf der SD-Leitung nur noch Nullen übertragen, solange bis das Signal WS von High auf Low wechselt.
- Die Periode von WS, auch Audio Frame genannt, entspricht der Abtastperiode ($1/\text{Abtastrate}$) des übertragenen Stereo-Signals. Im DTP2-Projekt wird eine Abtastrate von 48kHz verwendet.

3 Implementation der I2S-Audio-Blöcke

Die Implementation der I2S-Audio-Blöcke teilt sich auf in den Audio-Controller (audio_ctrl) und den I2S-Master (i2s_master) (siehe Abbildung 2.2)

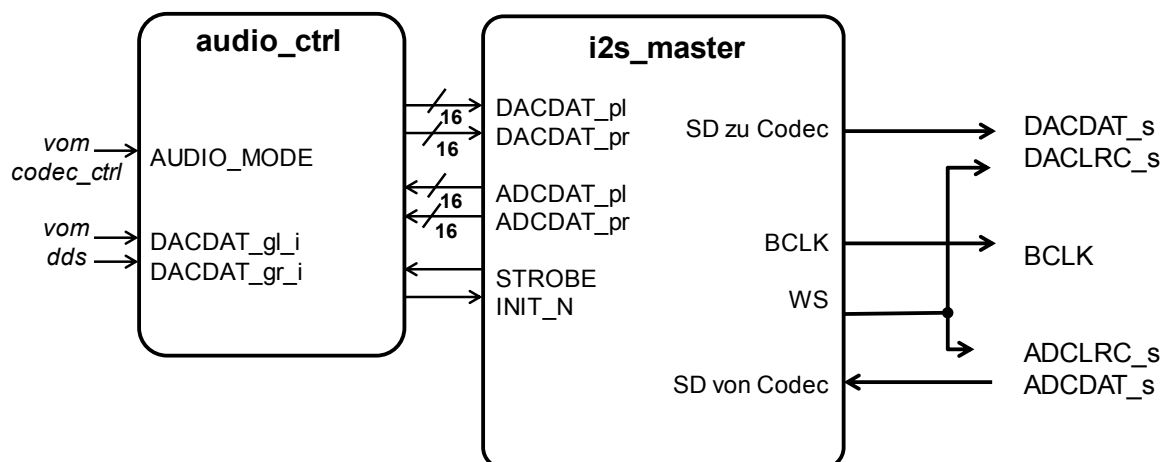


Abbildung 2.2: Entities der I2S Audio Blöcke

3.1 I2S Master (i2s_master)

Der Synthesizer des DTP2-Projektes soll intern mit parallelen 16-bit Signalen arbeiten. Der I2S-Master Block (siehe Abb. 1.2) wandelt serielle in parallele, sowie parallele in serielle Daten um und erzeugt die Signale BCLK und WS. Das SD-Signal *ADCDAT_s* ist der Eingang zum FPGA für Audio-Signale, die von den Codec-Line-Eingängen kommen. Auf der Parallel-Schnittstellen-Seite des I2S-Masters stehen zwei 16-bit Signale (links und rechts) vom AD-Wandler zur Verfügung *ADCDAT_p* sowie zwei Eingänge (links und rechts) zum DA-Wandler *DACDAT_p*. Der Ausgang *Strobe* ist ein Puls, der dem Synthesizer mitteilt, dass jetzt ein weiteres Audio-Wort an den DA-Wandler des Codecs gesendet wird.

Wenn der Eingang *init_n* low ist, wird der I2S-Master angehalten und die Register des I2S-Masters für die Übertragung initialisiert. Sobald *init_n* high wird, beginnt der I2S-Master serielle Daten an den Codec zu senden oder vom Codec zu empfangen. Der Eingang *rst_n_12M* ist der Master Reset, der alle Flip-Flops asynchron zurücksetzt.

Wichtig: Es gibt nur einen Masterclock im System, das ist *clk_12M*, mit dem alle Flip-Flops getaktet werden. Alle Ein- und Ausgänge müssen synchron zum *clk_12M* laufen. BCLK dient als Befähigungs-Signal zum Weiterschalten der Schieberegister.

3.2 Audio Controller (audio_ctrl)

Der Audio-Controller-Block wird benötigt, um zwischen Digital-Loop-Betrieb und Synthesizer-Betrieb umzuschalten und ist im Wesentlichen nur ein Multiplexer. Im Loop-Betrieb koppelt er das parallele Ausgangssignal des I2S-Masters auf das parallele Eingangssignal des I2S-Masters zurück. Der Audio-Controller-Block soll eine eigene Entity im Toplevel erhalten.

4 Blöcke des I2S-Masters

Abbildung 3.3 zeigt die Blöcke, aus welchen der Audio-Controller zusammengesetzt ist.

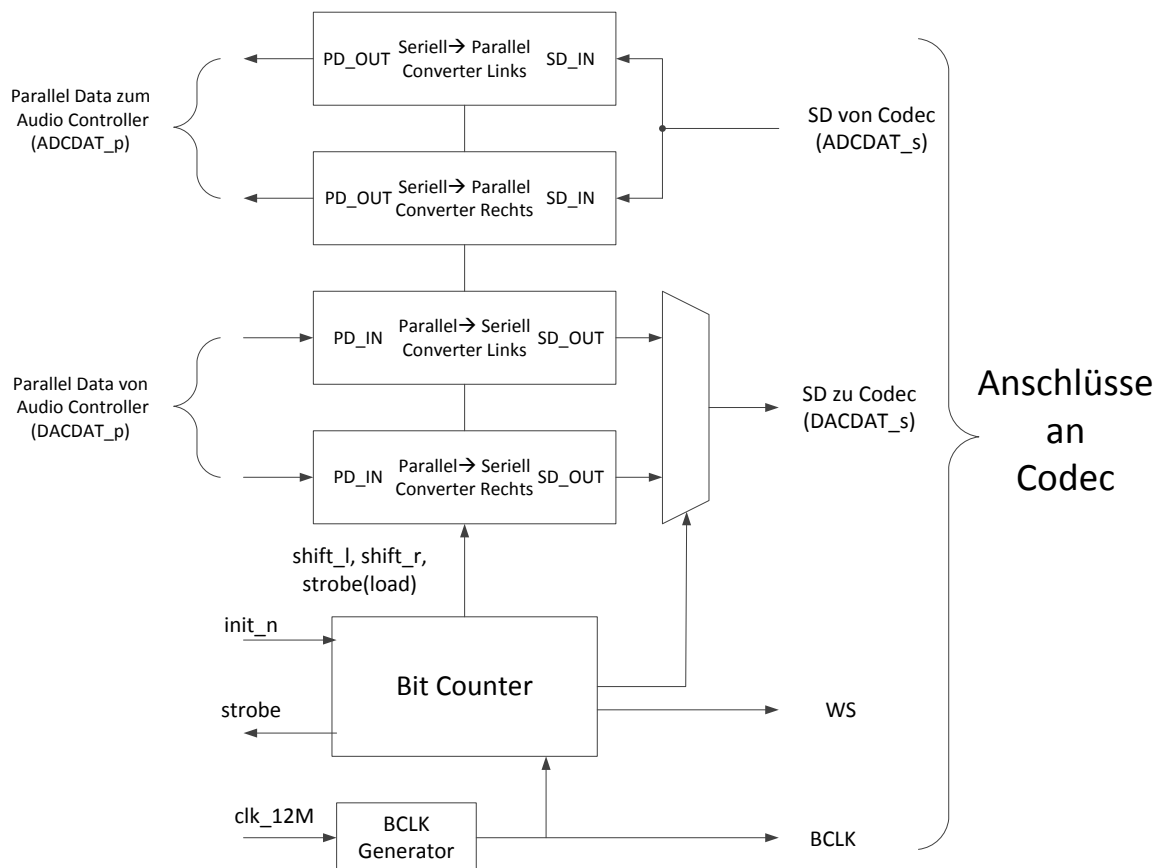


Abbildung 3.3: Abstrahiertes Blockdiagramm des I2S-Masters

4.1 Parallel -> Serial und Serial -> Parallel Wandler

Die Schaltungen, welche serielle in parallele SD-Daten und parallele in serielle Daten umwandeln, sind sehr ähnlich, sie unterscheiden sich nur in unwesentlichen Details. Aus diesem Grund kann für beide Funktionen ein gemeinsamer Schieberegister-Baustein entwickelt werden. Es ist der Gruppe überlassen, ob sie jeweils eine eigene Schaltung für seriell-zu-parallel-Umwandlung und parallel-zu-seriell-Umwandlung entwirft oder einen universellen parallel-zu-seriell-zu-parallel-Wandler entwickelt und diesen vier Mal instanziiert.

Eine universelle Parallel-Seriell-/Seriell-Parallel-Wandler-Entity verfügt über serielle und parallele Ein- und Ausgänge sowie über die folgenden Signale:

- clk_12M und rst_n_12M: Master Clock und asynchroner Reset für alle Flip Flops
- load: Wenn aktiv, lädt die Daten vom PD_IN Eingang in die Flip-Flops der Schieberegister
- shift: Wenn aktiv, schiebt die Daten an den SD_OUT Ausgang hinaus. **Hinweis:** Ist weder *load* noch *shift* aktiv, hält das Register die Daten im aktuellen Zustand.
- SD_OUT: sind die SD-Datenausgänge DACDAT_s zum Codec und werden zwischen dem linken und rechten Kanal gemultiplext (WS) (Umschalzeitpunkt beachten). Wird dieser Block als seriell-parallel-Wandler benutzt, bleibt dieser Ausgang offen (Im VHDL Port Map (signal => OPEN)).
- SD_IN: Sind die SD Dateneingänge, wenn man den Line Input des Codecs benutzen möchte. **Wichtig:** Wird dieser Block als parallel/seriell Wandler benutzt, so müssen diese Eingänge auf Low gelegt werden, damit nach der Übertragung der 16-Bits an den seriellen Ausgängen Nullen erscheinen.
- PD_IN: Paralleler Eingang, wenn man Daten zum Kopfhörer senden möchte. Wird dieser Block als seriell-parallel-Wandler benutzt, legt man diese Eingänge auf Low.
- PD_OUT: Paralleler Ausgang wenn man den Line Input des Codecs benutzen möchte. Wird dieser Block als parallel-seriell-Wandler benutzt, bleibt dieser Ausgang offen (Im VHDL Port Map (signal => OPEN)).
- BCLK: Qualifizierer für das Schieben. Nur wenn BCLK high ist, schieben die Register ihre Bits weiter.

Abbildung 1.4 zeigt die Ein- und Ausgangssignale eines universellen Schieberegisters.

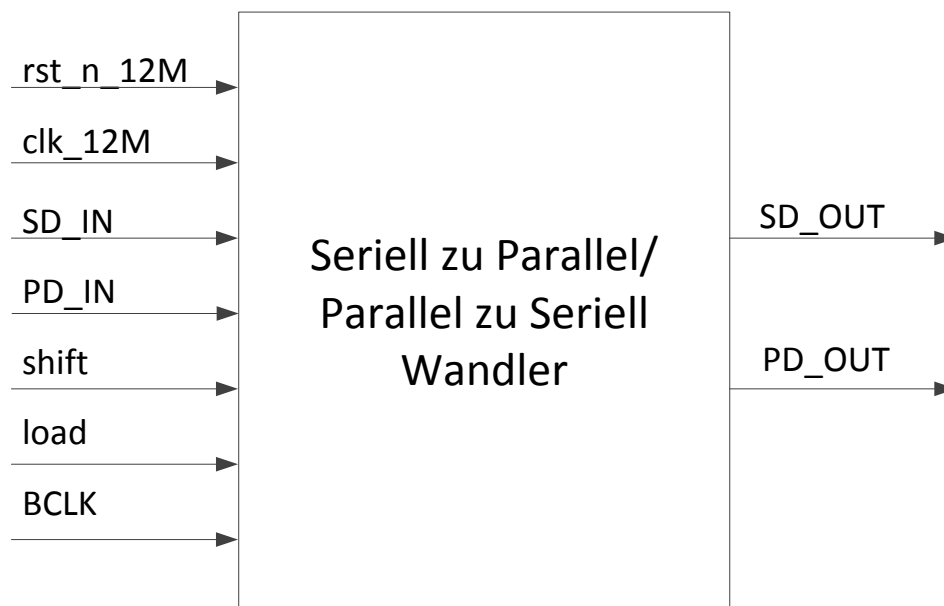


Abbildung 4.4: Anschlussdiagramm des universellen parallel-seriell-Wandlers

4.2 Bit Counter und BCLK Generator

Die Steuersignale für die Schieberegister werden durch einen umlaufenden Zähler und einen Decoder generiert. Abb. 1.5 zeigt das Zeitverlaufdiagramm der Schieberegisteransteuerung.

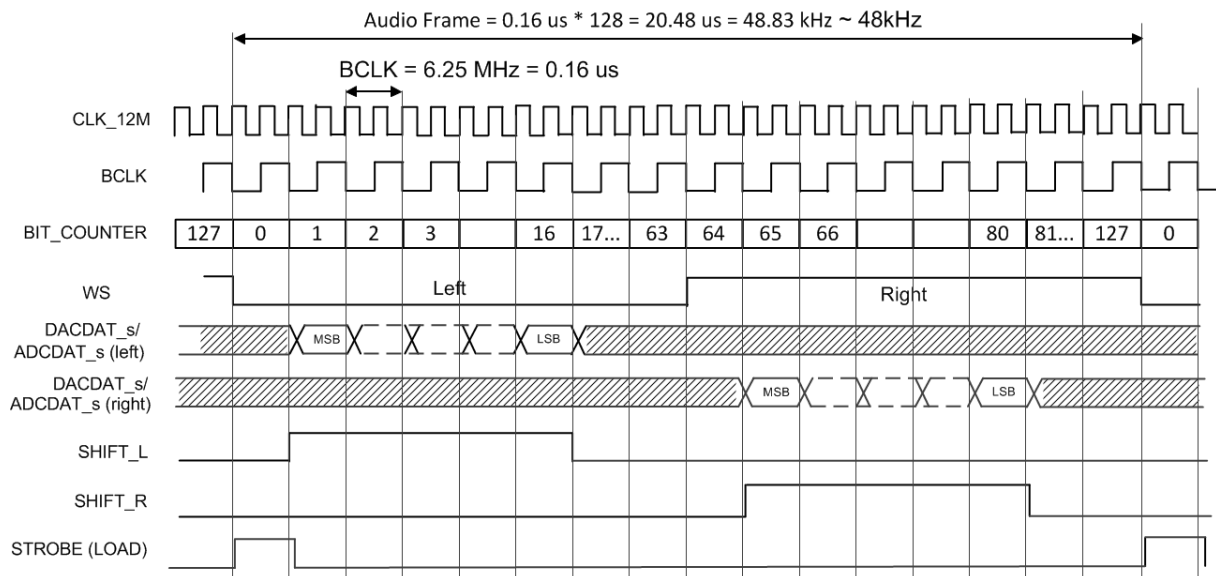


Abbildung 5.5: Zeitverlaufdiagramm der Schieberegisteransteuerung

Das Signal BCLK, welches als I2S-Bit-Clock für die Codecs dient, soll gleichzeitig als Qualifizierer für die Schieberegister verwendet werden. Das heisst, die Schieberegister schieben nur dann weiter, wenn BCLK high ist. $BCLK = clk_12M/2 = 12.5 \text{ MHz}/2 = 6.25 \text{ MHz}$. $TBCLK = 0.16 \mu s$. Der Zähler wird mit BCLK inkrementiert, aber mit clk_12M getaktet. BCLK qualifiziert jedoch das Hochzählen des Zählers, d.h. der Zähler zählt nur dann weiter, wenn BCLK high ist. Der BCLK darf nie verwendet werden, um Takteingänge von DFFs zu versorgen, da sonst das Timing der daran angeschlossenen DFF verletzt werden würde.

Für eine Abtastrate von 48 kHz bedeutet dies, dass innerhalb von $20.48 \mu s$ 2 Kanäle übertragen werden. Bei einer BCLK-Periode von $0.16 \mu s$ ergeben sich für den Zähler 128 Schritte, um die zwei Audio-Wörter (links und rechts) über die I2S-Schnittstelle zu senden und gleichzeitig zu empfangen.

Der Decoder in Abb. 1.6 dekodiert die Signale, um die Schieberegister, den WS, strobe etc. zu erzeugen.

Um den BCLK zu erzeugen, wird im BCLK-Generator in Abb. 1.3 die clk_12M durch 2 geteilt. Wenn das Signal $init_n$ low ist, soll der BCLK-Generator gestoppt werden und zwar so, dass BCLK low ist.

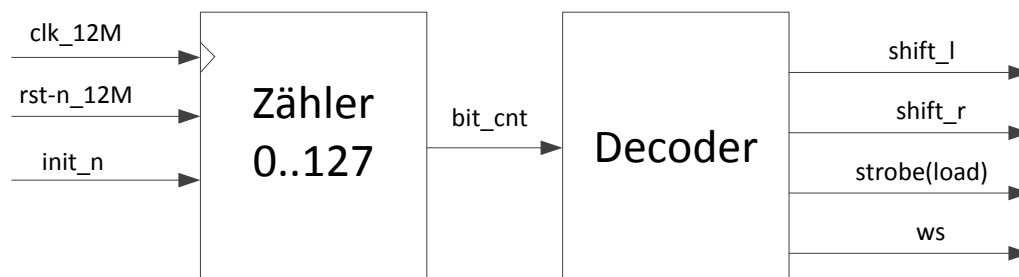


Abbildung 6.6: Aufbau des Controllers mit Zähler 0-127 und Decoder