

APPLICATION NOTE

AN2456

IEEE 1394 Audio Applications with
PDI1394L41/40

Author: Paul Schaefer

2001 Jul 13

IEEE 1394 audio applications with PDI1394L41/L40

AN2456

1. INTRODUCTION

1.1. Purpose and Scope

This document describes 1394 audio applications using the Philips 1394 link chips PDI1394L40 and PDI1394L41. The target audience is engineers that are using Philips Semiconductors 1394 silicon for digital audio designs.

1.2. Glossary

ADC	Analog to Digital Converter
AUDIO SAMPLE	A discrete sample of a single channel data
AM824	Audio/Music data format with an 8-bit header and 24 bits of audio
CIP	Common Isochronous Packet header defined by IEC61883-1
Cycle Timer Register	Link Cycle Time
DAC	Digital to Analog Converter
DBC	Data Block Counter
FDF	Format Dependent Field of CIP header
I ² S	Inter Integrated Circuit Serial audio transmission
I/O	Input/Output
MIDI	Musical Instrument Digital Interface
PCR	Plug Control Register
PLL	Phase Locked Loop
SPDIF	Sony/Philips Data Input Format
USB:	Universal Serial Bus

1.3. References

- [1] IEEE1394-1995 Standard for a High Performance Serial Bus
- [2] IEC61883 part 1 Digital Interface for Consumer Audio/Video Equipment
- [3] IEC61883 part 6 Specification for Audio and Music Data Transmission
- [4] Enhancement to Audio and Music Data Transmission Protocol 1.0, 1394 TA document 1999014
- [5] AV/C Digital Interface Command Set General Specification, version 3.0, April 1998, 1394 Trade Association
- [6] ISO/IEC 13213-1994 (ANSI/IEEE1212-1994): Information Technology-Microprocessor Systems-Control and Status Register (CSR) architecture for Microcomputer Buses.
- [7] AV/C Audio Subunit Specification 1.0, 1394 TA document 1999008
- [8] AES3-1992: Serial transmission format for two channel linearly represented digital audio data. Audio Engineering Society (ANSI S4.40-1992).

2. OVERVIEW

The IEEE 1394 bus [R1] is well suited for transferring audio and control information. The 1394 trade association has published two standards for the upper layer protocols using 1394 for audio streaming and control:

1. The Audio and Music Data Transmission protocol, also known as IEC61883-6 [R3], which we will refer to in this document as the A&M Protocol V1.0.
2. The Enhancement to Audio and Music Data Transmission Protocol, V1.0 [R4].

The purpose of this application note is to describe some of the special considerations when designing the PDI1394L40/L41 into an audio system compliant with these documents.

3. HARDWARE CONFIGURATION

Figure 1 shows a block diagram of the Philips Semiconductors 1394 audio demonstration board centered around the Philips PDI1394L40 link chip.

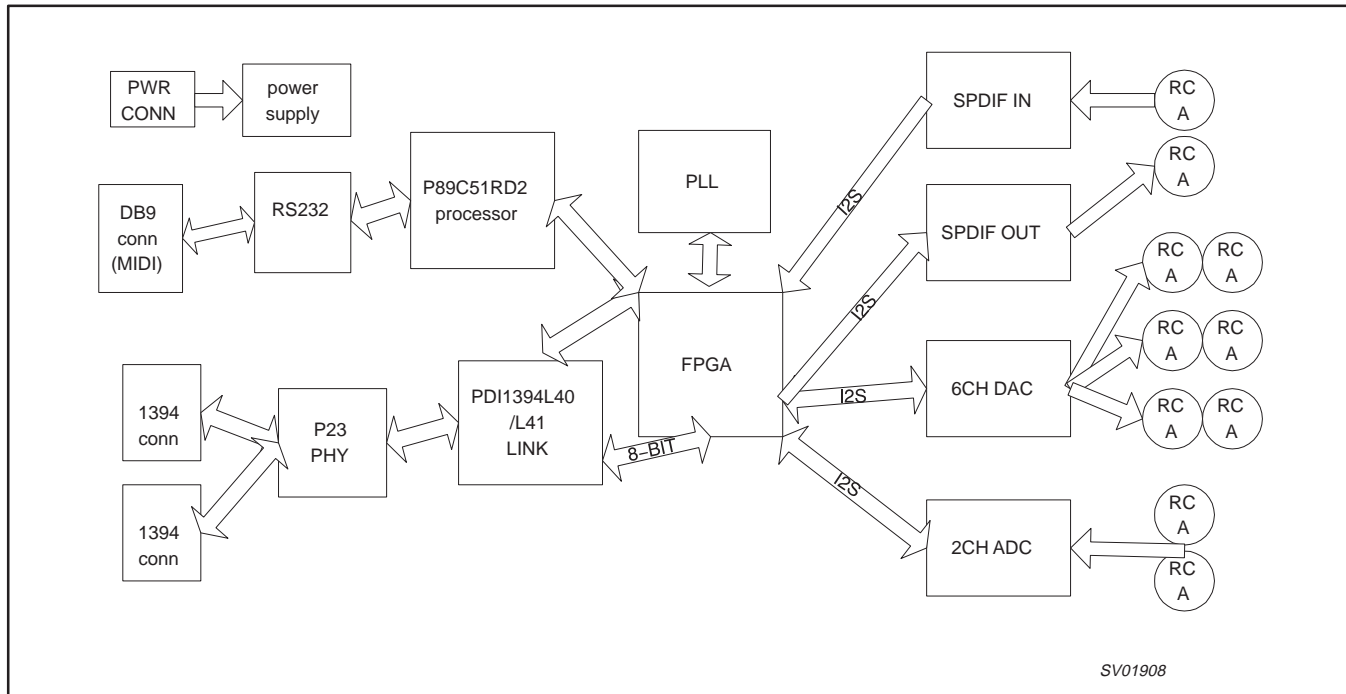
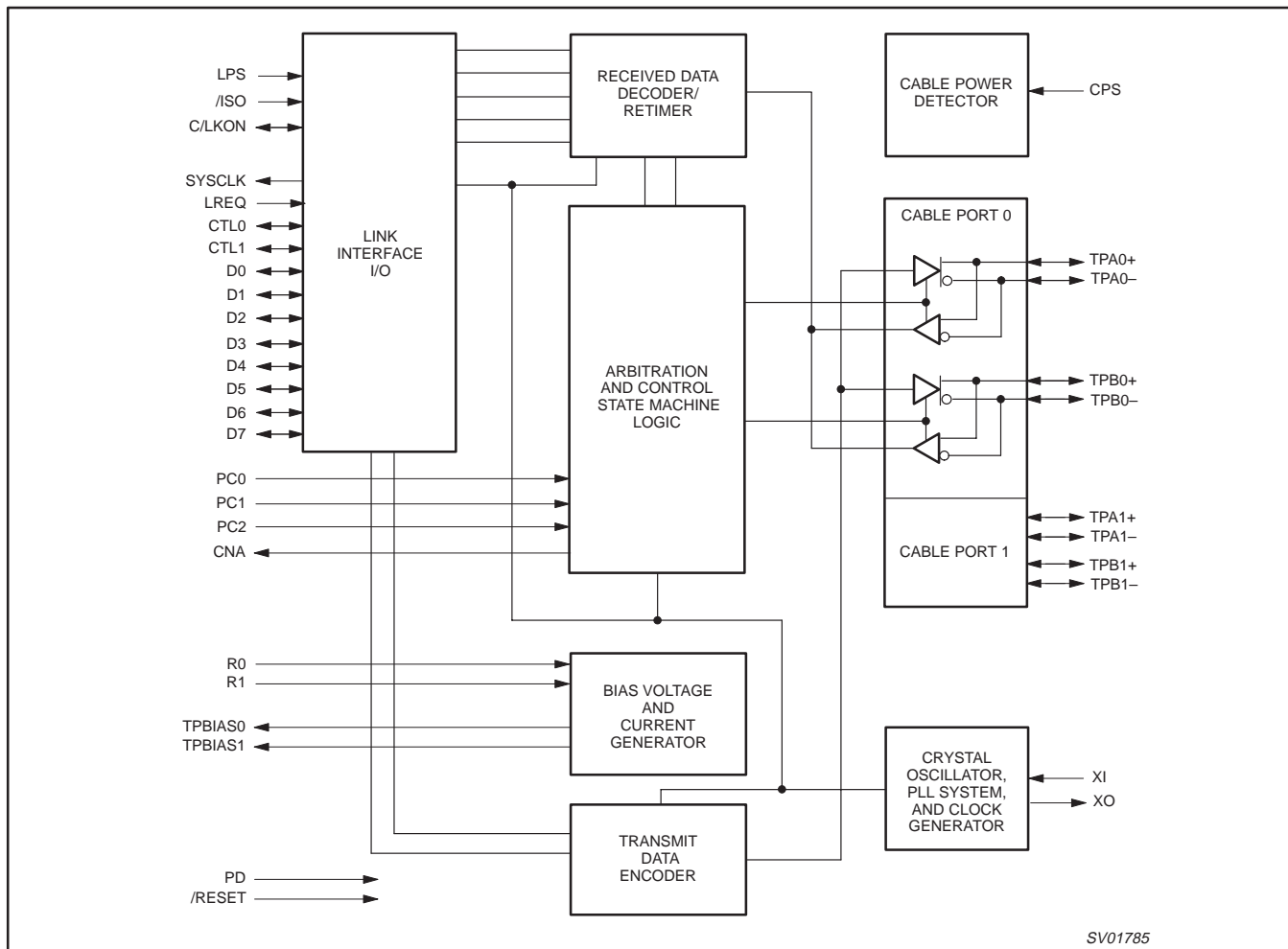


Figure 1. Philips 1394 Audio Board (Rev 2.0) Block Diagram

The datapath between the link chip and the FPGA is 8 bit (parallel), and the datapath from the link chip to/from the ADC and DAC parts is I2S (serial). The FPGA converts between the parallel to serial formats and inserts/removes audio headers.

The PLL is used to recover the audio clocks from time stamp data sent with the audio data over 1394.

4. PDI1394P23 BLOCK DIAGRAM



5. CONSIDERATIONS FOR AUDIO TRANSMIT

The A/M Protocol V1.0 supports several modes of packet transfer over 1394:

1. Blocking mode with CIP-only cycles for no-data cycles
2. Blocking mode with CIP and garbage data for no-data cycles
3. Non-blocking mode

In blocking mode for a given 1394 bus cycle the transmitter will only send audio samples in blocks that have a number of samples equal to the SYT_INTERVAL defined for the audio sample rate. For audio with sampling rates from 32 KHz to 48 KHz the SYT_INTERVAL is 8 (ref #3), so in this case for any given bus cycle the transmitter will either send 8 audio samples or none. The empty bus cycles contain either just an 8-byte CIP header (defined below), or else the CIP header plus fill data equal in size to 8 audio samples (garbage data).

In non-blocking mode the transmitter sends as many audio samples as it has in its buffer on every bus cycle, so the samples are transmitted in a fairly constant rate without any empty bus cycles.

CIP stands for Common Isochronous Packet. This 8-byte header, defined in IEC61883-1 [R2] consists of the first 8 bytes of the data payload of the isochronous packet, and contains information about the audio or video data in

IEEE 1394 audio applications with PDI1394L41/L40

AN2456

the remainder of the packet. Two of the fields of the CIP header are of special interest to us: the SYT time stamp, and the data block counter (DBC).

The SYT time stamp is a 2-byte value that is compared by the receiver to the lower two bytes of a local cycle timer register of the Link Layer Controller. When these two bytes are equal the SYT Time Stamp expires. The purpose of the SYT is to allow a low frequency free running clock on the transmitter to be reproduced (with a fixed delay) at the receiver. Since the 1394 bus cycles occur at a frequency of 8 KHz, and only one SYT Time Stamp can be sent per channel per bus cycle, the SYT Time Stamp recurrence rate is limited to a maximum frequency 8 KHz.

Audio or video data compliant to IEC61883–1 is broken into AV Application packets of a fixed size, and these AV packets can be further divided into data blocks. Each AV packet consists of 1, 2, 4 or 8 data blocks. The CIP header indicates the size of the AV packet and how many data blocks make up the AV packet. For the A/M Protocol V1.0 the AV packet is not divided so the AV packet is equal to one data block.

The CIP header also has a data block counter (DBC) which is a 1 byte counter that points to the first data block of the packet included in the current bus cycle. This is used by the receiver to keep track of how many data blocks have been received.

The A/M Protocol V1.0 defines:

- 1 AV packet = 1 data block = 1 audio sample

This results in the somewhat unusual situation where any given 1394 bus cycle will normally carry multiple AV packets.

The A/M Protocol V1.0 also defines that when an SYT time stamp is sent with a packet, it corresponds to a particular audio sample in that packet, namely the sample that is a multiple of the SYT_INTERVAL.

For example if the SYT interval is 8 and we receive an isoch packet with a valid SYT and the DBC is equal to 0x06 the time stamp does not correspond to the first audio sample in the isoch packet. In this case the time stamp corresponds to the third sample because the DBC for the third sample will be a multiple of 8. The A/M Protocol V1.0 defines that this audio sample should be presented at the receiver at the time that its SYT expires.

The reason this is important for transmit is because it means that we have to make sure that the SYT gets sent in the same isoch packet that contains the audio sample that it points to.

When programming the PDI1394L40/L41 for audio transmit the packing mode (PM) needs to be set to b01 for blocking mode, and the AUDIO mode bit should be set. Both of these settings are found in the ITXPKCTL link register at offset 0x020. If non-blocking mode is used then the time stamps will not align with the correct audio samples, so make sure that blocking mode is enabled.

The SYT time stamp is generated from a clock input to the AVFSYNC pin of the link chip. The FPGA generates this clock at a rate that is a factor of SYT_INTERVAL slower than the audio sample rate. In the case of 32 KHz–48 KHz audio the AVFSYNC clock is 8 times slower than the audio, and for 96 KHz audio the factor is 16. So in the case of 48 KHz audio the SYT_INTERVAL is 8 and the AVFSYNC clock is 6 KHz. This clock must be phase locked to the audio clock. The ITXHQ1 and ITXHQ2 registers of the PDI1394L40/41 need to be initialized to the correct values for the audio sample rate that is being transmitted, but then the Philips link chip automatically creates and appends the CIP headers to the isochronous packets.

Figure 2 shows a 1394 capture of stereo audio transmitted from the Philips link chip in blocking mode according to the A/M Protocol V1.0 in the AM824 format.

IEEE 1394 audio applications with PDI1394L41/L40

AN2456

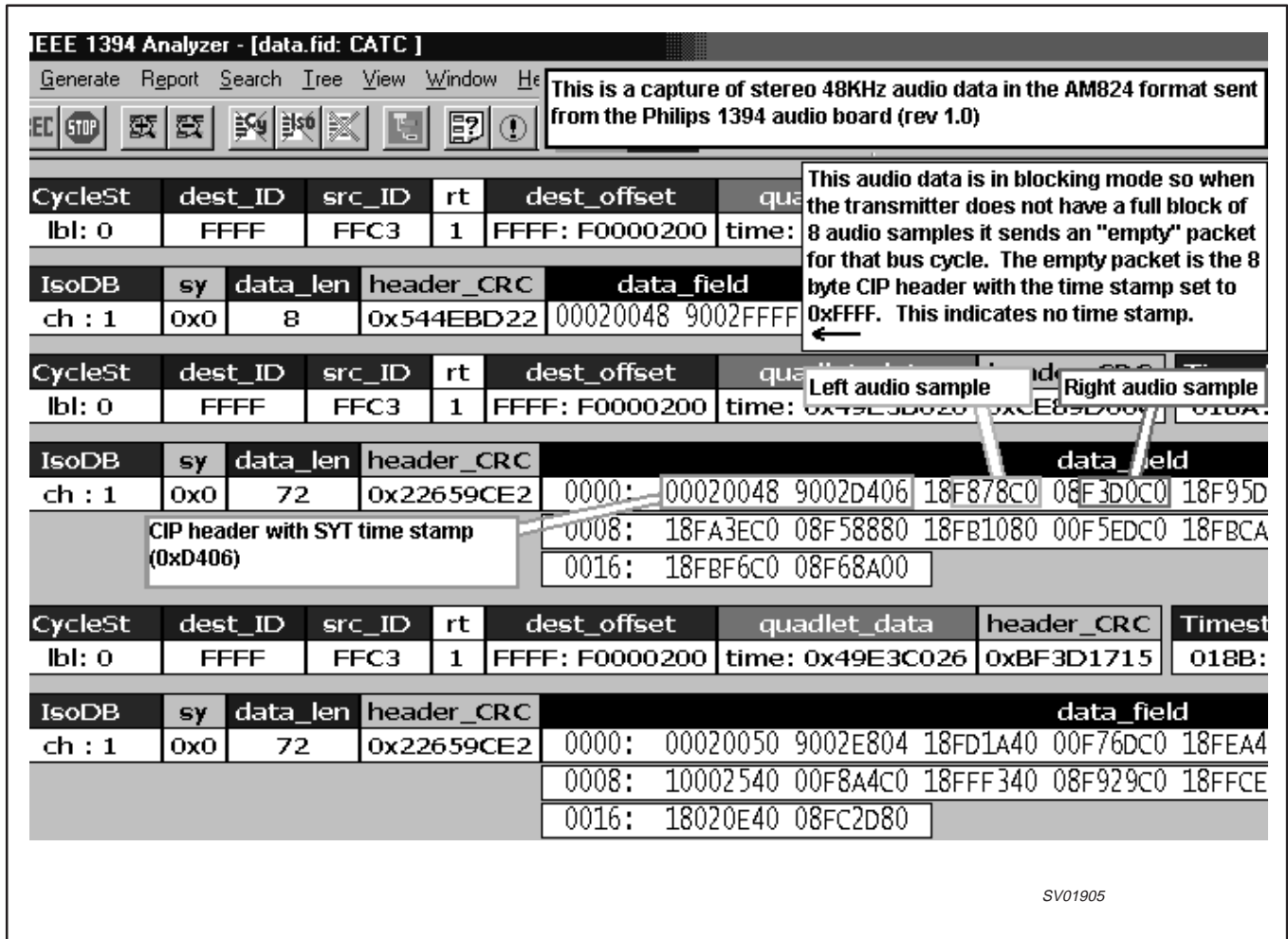


Figure 2. 1394 Capture of 48 KHz Audio Data in Blocking Mode

In the example in Figure 2 the first full isochronous packet has a datablock counter (DBC) value of 0x48 and the packet contains 8 audio samples. The next isochronous packet has a datablock counter value of 0x50, which is 0x8 more than the previous counter, as expected. In both cases the datablock counter is divisible by 8, so the SYT corresponds to the first audio sample pair in the packet.

The example in Figure 2 is for audio data in the AM824 format as defined in the A/M Protocol V1.0. The AM824 format consists of 32 bits for each audio sample where the most significant 8 bits are a header defined by the IEC958 standard [R8], and the lower 24 bits are for the audio sample. On the Philips 1394 audio board when the audio source is the ADC the FPGA automatically generates the 8-bit header before passing the incoming audio on to the link chip. If board receives audio from a SPDIF receiver then the 8-bit header information needs to be passed from the SPDIF chip to the FPGA before the data is sent to the link chip.

The AM824 format is the most commonly used format of those defined in the A/M Protocol V1.0. The Philips PDI1394L40/L41 is capable of supporting the various audio formats defined in the A/M Protocol, but currently the demonstration board only has software support for the AM824 format.

6. CONSIDERATIONS FOR AUDIO RECEIVE

Once again we consider the three modes of packet transfer defined in the A/M Protocol V1.0:

1. Blocking mode with CIP-only cycles for no-data cycles
2. Blocking mode with CIP and garbage data for no-data cycles
3. Non-blocking mode

Modes 1 and 3 work great with the PDI1394L40/L41 because of the chip's ability to automatically remove the CIP headers and reassemble the AV packets from the isochronous stream. Mode 2 is currently not supported by the PDI1394L40/L41, but we are currently not aware of any 1394 audio implementations that use the garbage data format, so this limitation should not cause the A/M Protocol V1.0 compatibility problems in the field.

When the audio is received from 1394 the audio clocks need to be regenerated from the SYT time stamps sent in the CIP header portion of the data payload of the isochronous packets. The PDI1394L40/L41 will regenerate a clock from the SYT time stamps by means of the signal on the AVFSYNC pin. This signal can be used as a reference for a PLL to generate the audio clocks. The PLL can remove much of the jitter inherent in the AVFSYNC signal resulting in a low jitter audio clock. The incoming audio samples need to be buffered to allow the samples to be played back at the rate of the audio clock independent of the rate that the audio samples arrive from 1394. The PDI1394L40/L41 has a receive buffer of adjustable size that can be used to perform this buffering function.

The L40/L41 are limited to buffering 64 audio samples because this is the limit of how many AV packets the link will hold before reporting a status queue overflow error.

An AM824 audio sample is 4 bytes = 1 quad, so if you multiply the number of channels that you are sending by 64 then you get the number of quads needed to buffer.

For example to send 6 channel audio, the recommended FIFO size is:

- $6\text{ch} \times 1\text{quad/ch} \times 64 = 384\text{quads} = 1536\text{bytes}$

The PDI1394L40/L41 can deliver the received audio samples as soon as they have arrived from the 1394 bus, but on the Philips 1394 audio demo board we allow the link chip to hold about 32 audio samples in its isochronous receive buffer before we start to clock them out of the link chip.

The purpose in buffering the samples in the link FIFO is to allow the PLL-recovered audio clocks to clock out the audio samples at an even rate independent of the rate that they arrive from the 1394 isochronous packets.

When a new audio playback stream arrives from 1394 the PLL of the audio system has to synchronize to the rate of the incoming time stamps to synchronize the playback with the timestamps. The audio samples are discarded temporarily by asserting AVREADY on the link AV interface and discarding the audio samples as we wait for the PLL to lock to the new SYT time stamps. Once the PLL is locked we assert AVREADY so the link begins to store samples in its internal AV FIFO. We then wait for a time equivalent to 32 audio samples, which depends on the audio sample rate. For example, with 44.1KHz audio we would wait for:

- $32 \times (1\text{ audio sample period}) = 32 \times 22.7\ \mu\text{S} = 725\ \mu\text{S}$

This lets the FIFO fill up with roughly 32 audio samples. Then we start clocking out samples at the audio rate, e.g., 44.1KHz.

The PLL clocks the samples out of the link at the same average rate that they are arriving from 1394, but the data arrives from 1394 in bursts, so the FIFO level is going to move around a lot during normal operation.

IEEE 1394 audio applications with PDI1394L41/L40

AN2456

The FIFO can hold a maximum of 64 audio samples, or +/- 32 audio samples, so we need to make sure the PLL has a relatively quick response so that small changes in the audio rate will not result in a FIFO overflow. But the PLL response should not be too fast because this would result in too much sample jitter in the playback audio.

Figure 3 shows a logic analyzer plot of the AV port signals for 1394 audio being received by the Philips 1394 audio board.

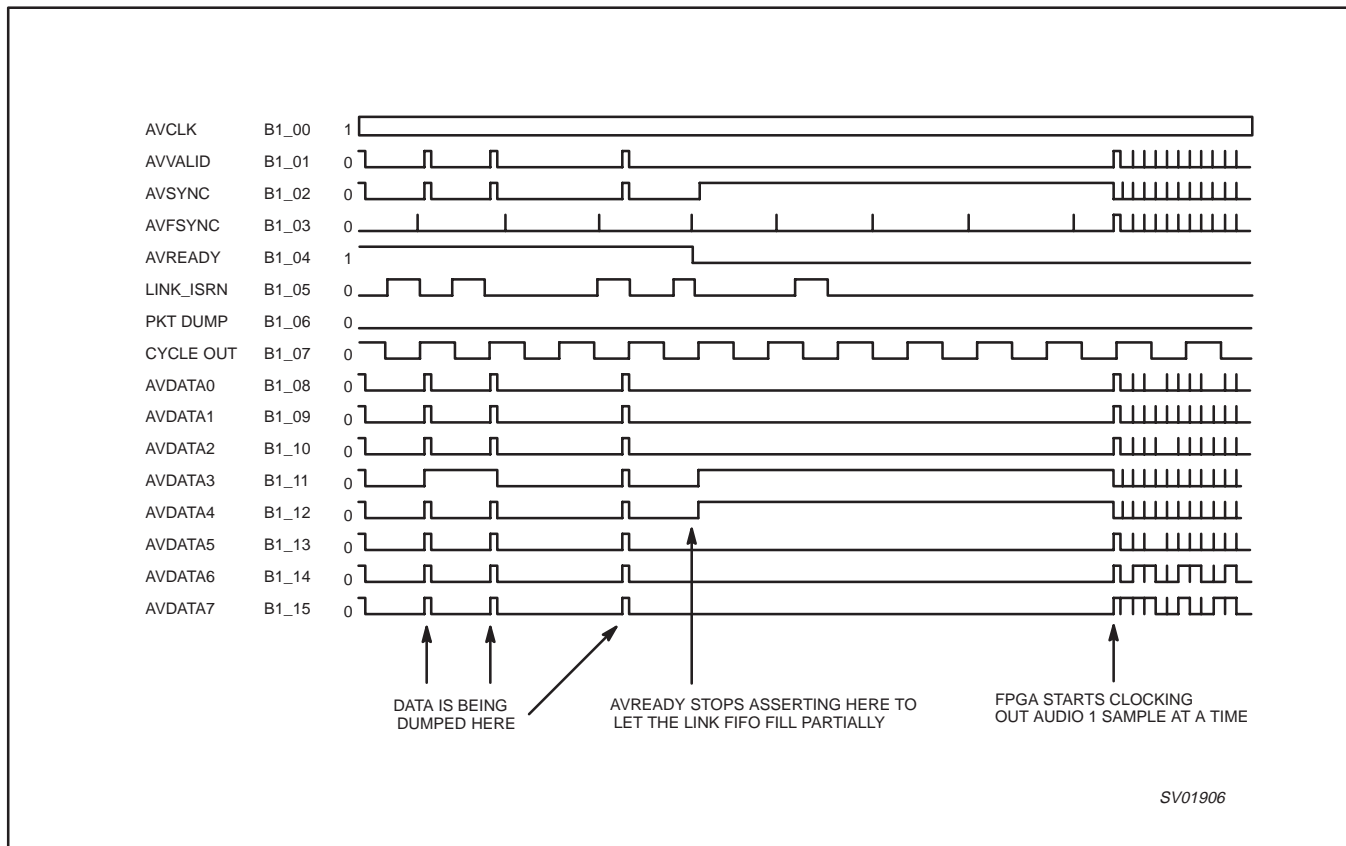


Figure 3. Logic Analyzer AV Port Signals for Receive of 1394 Audio

In this case the audio board is making a transition from dumping the audio packets to clocking them out at the audio rate. The reason that the audio board dumps the packets initially is to allow the PLL time to lock to the audio clock signals from the AVSYNC pin. Once the PLL has locked the processor instructs the FPGA to stop asserting AVREADY for a fixed period to allow the link FIFO to fill with audio samples. Once the FIFO is sufficiently full the processor instructs the FPGA to start clocking out audio samples one sample at a time at the audio rate.

Figure 4 shows the AV signals for a single sample of audio being clocked out of the link chip. The AVCLK is free running (source?) and AVVALID is used to regulate the data output from the link.

IEEE 1394 audio applications with PDI1394L41/L40

AN2456

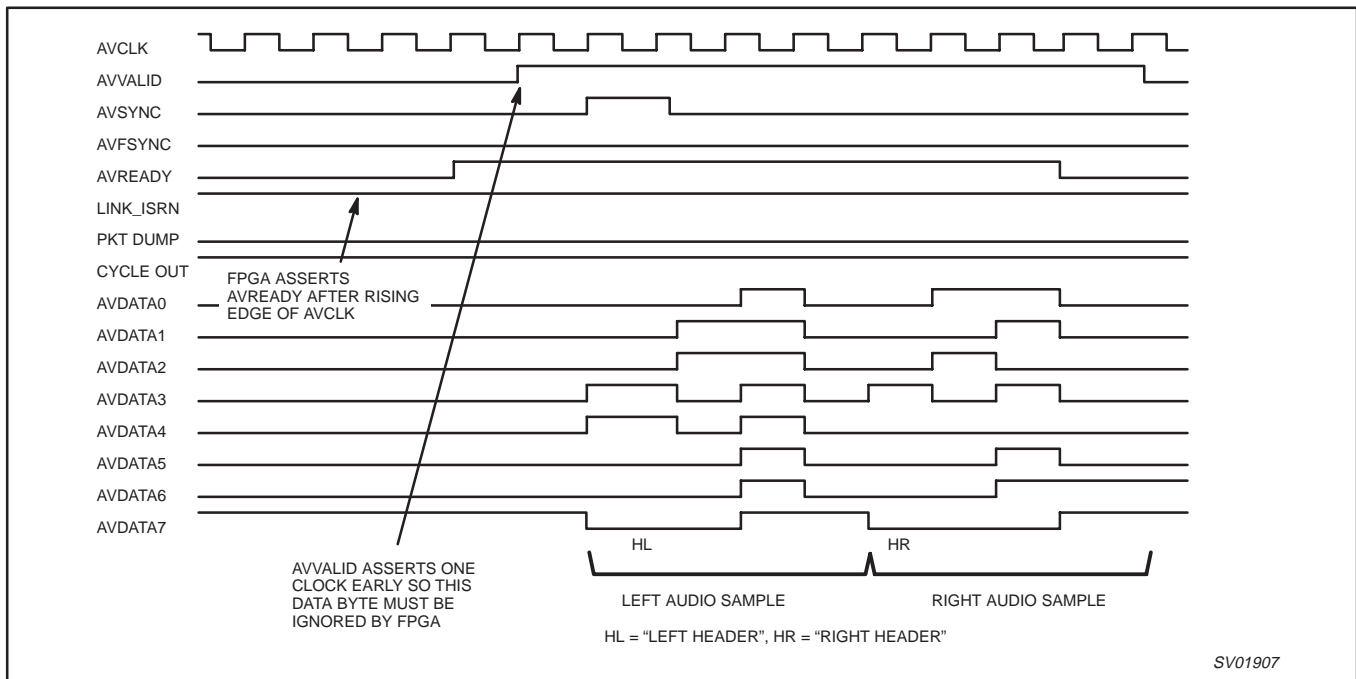


Figure 4. Logic Analyzer AV Port Signals for one stereo audio sample

The scenario described above results in a cost effective 1394 audio design because the buffering for the received audio packets is done internal to the PDI1394L40/L41.

The timing for the AVREADY signal is very critical during the initial audio synchronization because there needs to be correspondence between the SYT Time Stamp expiration time and the delivery of the received audio sample that the SYT Time Stamp points to. The PDI1394L40/41 strips off the CIP headers from the incoming isochronous packets and uses the CIP information to reassemble the AV packets contained in the isochronous packets. This CIP header removal is normally beneficial, but it causes a problem during audio synchronization as it makes it difficult to tell which audio sample a particular SYT is pointing to. This limitation will be addressed in a future version of the Philips link silicon, but in the mean time we have a suggested work-around:

During the packet dumping period all audio packets are discarded. As soon as the PLL has locked to the audio the processor can instruct the FPGA to interrupt it on every bus cycle. When the interrupt occurs for each bus cycle the SYT field is read by the processor by accessing the CIP header for that bus cycle by reading the IRXHQ2 register of the link chip. When an SYT with the value 0xFFFF is detected this corresponds to a bus cycle that contains no audio samples divisible by the SYT_INTERVAL, and it also means that the next bus cycle will have an SYT that points to an audio sample at or very close to the beginning of the its corresponding packet. An appropriate CPU/Clock must be able to service 8000 interrupts per second.

At the beginning of the next bus cycle after the 0xFFFF time stamp detection the FPGA stops asserting AVREADY to allow the link to buffer the 1394 audio packets. When the processor is interrupted for the next bus cycle the SYT field is read and stored.

At this point the processor knows the value of the SYT time stamp that corresponds to the first (or nearly the first) audio sample in its buffer. At this point there are two possible scenarios:

1. The processor can calculate a cycle time that is slightly less than the SYT expiration time. When the processor sees that the cycle timer is approaching the value of the SYT expiration it can notify the FPGA to

start clocking audio samples out of the AV port of the link when the FPGA sees the next rising edge of the AVFSYNC signal.

2. The processor can calculate the FPGA notify time as in 1) but also add to this time a fixed number of SYT time intervals depending on how many samples we wish to have buffered in the FIFO of the PDI1394L40/L41. The result is similar to scenario 1), but in this case the link will carry more audio samples internal to its buffer and the output audio will have an added delay from the transmitter.

In both scenarios once the audio is being clocked out by the FPGA the processor no longer needs to be interrupted on every bus cycle unless the link chip reports an error in the sequence of the DBC of the incoming isochronous stream. If a DBC error occurs then the processor has to repeat the synchronization procedure.

The advantage to scenario 1) is that the audio samples will be presented from the receiver to the application layer at or very close to the expiration time of the SYT time stamp that they correspond to, which is a requirement of A/M Protocol V1.0. The disadvantage to scenario 1) is that the audio samples will need to be buffered outside of the link chip, which adds cost to the system.

Scenario 2 provides a sample-accurate playback time with a fixed delay from the presentation time indicated in the SYT Field. This is not strictly compliant to the A/M Protocol V1.0, but it results in a more cost effective design than scenario 1. It is up to the designer to decide which method is appropriate for their product.

The Philips 1394 audio board uses scenario 2 to take advantage of the internal isochronous FIFO of the link chip for reduced cost. Currently only the AM824 format is supported by the board firmware, but other formats defined by the A/M Protocol V1.0 can easily be supported with modifications to the FPGA. The format type of a received audio stream is indicated in the FDF field, which is accessible in the link register IRXHQ2.

7. CONSIDERATIONS FOR MIDI TRANSMIT AND RECEIVE

The A/M Protocol V1.0 defines support a single MIDI stream per isochronous channel where the MIDI data is sent in the AM824 format with a special 8-bit header (0x80–0x83) to identify the data as MIDI rather than audio. The PDI1394L40/41 can transmit and receive such a MIDI stream using the FPGA to transfer the MIDI message between the link and the processor. The processor can then include serial MIDI ports to allow standard MIDI devices to communicate via 1394.

The Enhancement to A/M Protocol [R4] defines a new method of MIDI transmission that multiplexes multiple MIDI streams on a single isochronous channel using the DBC distinguish the streams. Since the PDI1394L40/41 strips off the CIP header from the isochronous stream the DBC is no longer available to the application layer, so it is currently not possible to support this method of MIDI multiplexing with the current silicon. A future version of Philips link silicon will add this support.

IEEE 1394 audio applications with PDI1394L41/L40

AN2456

Definitions

Short-form specification — The data in a short-form specification is extracted from a full data sheet with the same type number and title. For detailed information see the relevant data sheet or data handbook.

Limiting values definition — Limiting values given are in accordance with the Absolute Maximum Rating System (IEC 134). Stress above one or more of the limiting values may cause permanent damage to the device. These are stress ratings only and operation of the device at these or at any other conditions above those given in the Characteristics sections of the specification is not implied. Exposure to limiting values for extended periods may affect device reliability.

Application information — Applications that are described herein for any of these products are for illustrative purposes only. Philips Semiconductors make no representation or warranty that such applications will be suitable for the specified use without further testing or modification.

Disclaimers

Life support — These products are not designed for use in life support appliances, devices or systems where malfunction of these products can reasonably be expected to result in personal injury. Philips Semiconductors customers using or selling these products for use in such applications do so at their own risk and agree to fully indemnify Philips Semiconductors for any damages resulting from such application.

Right to make changes — Philips Semiconductors reserves the right to make changes, without notice, in the products, including circuits, standard cells, and/or software, described or contained herein in order to improve design and/or performance. Philips Semiconductors assumes no responsibility or liability for the use of any of these products, conveys no license or title under any patent, copyright, or mask work right to these products, and makes no representations or warranties that these products are free from patent, copyright, or mask work right infringement, unless otherwise specified.

Philips Semiconductors
811 East Arques Avenue
P.O. Box 3409
Sunnyvale, California 94088-3409
Telephone 800-234-7381

© Copyright Philips Electronics North America Corporation 2001
All rights reserved. Printed in U.S.A.

Date of release: 07-01

Document order number:

9397 750 08561

Let's make things better.