IEEE 1394 Architecture

The **IEEE Standard 1394-1995** standard for the High Performance Serial Bus defines a serial data transfer protocol and interconnection system. The capabilities of the 1394 bus are designed to support a variety of high-end digital audio/video applications.

The IEEE's Microcomputer Standards Committee released the IEEE 1394-1995 standard in fall 1995, based on Apple Computer's original **FireWire™** bus (originally Apple trademark now 1394 Trade Association), which was intended as a low-cost replacement for the SCSI bus. Sony created its own trademark for 1394 called **iLINK.**

The **1394 Trade Association** was founded in 1994 to support the development of computer and consumer electronics systems that can be easily connected with each other via a single serial multimedia link, the 1394 bus. The serial bus follows the IEEE 1212 (IEC 13213) Control and Status Register Architecture (CSR) for 64 bit addressing, where the upper 16 bits of each address represent the node_ID, providing address space for up to 64,000 nodes in a system (Fig.1).

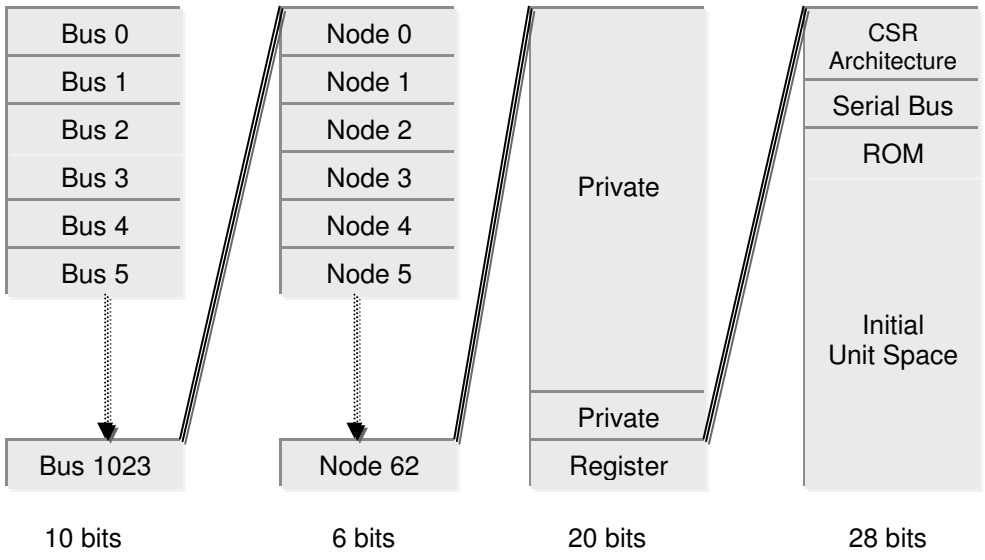

Figure 1- Bus Addressing

The node_ID is divided into the 10 bits bus_ID and the 6 bits physical_ID, providing **1023 buses**, each with **63 addressable nodes**. The lower bits address the node configuration ROM (to identify the node) and a standard set of control registers (CSR).

Up to **16 nodes** can be daisy-chained (cable hops) with standard cables up to **4.5 m** in length for a total standard cable length of 72 m, thus the term finite branches.

The 1394 cable bus is a non-cyclic (loops are forbidden) network with finite branches, which means that **tree and daisy-chain** topologies are supported.

After a bus reset (on power up or whenever a node is added or removed from the system) the bus is initialized and a tree identification (Tree_ID) process translates the general topology into a tree topology with branches and leaves (Fig.2). One node is designated as a root (highest priority). The self-identification (Self_ID) process then assigns a physical_ID to each node. 1394 is **plug-and-play,** no device ID discrete switches are required and **hot plugging** of nodes is supported.

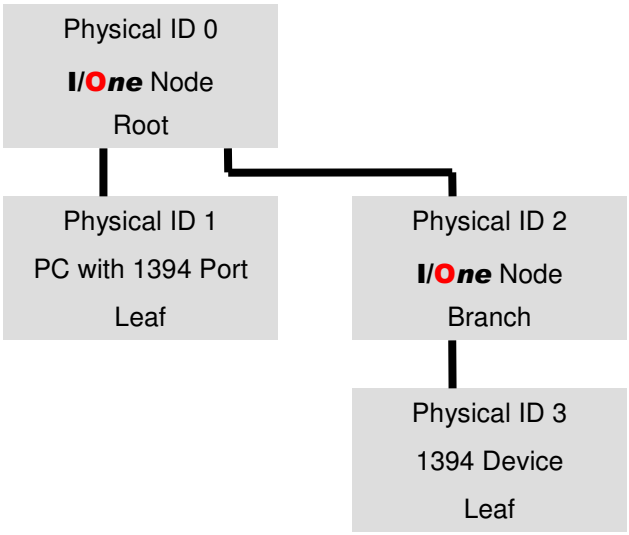

Figure 2- Tree After bus reset

Physical, Link, and Transaction Layers

The three-stacked layers shown in figure 3 implement the 1394 protocol. The **transaction layer** implements the request-response protocol required to perform the bus transactions and conform to the IEEE 1212 CSR Architecture (the operations of read, write and lock). The **link layer** supplies an acknowledged datagram (a one-way data transfer with request confirmation) service to the transaction layer. The link layer handles addressing, data checking and data framing for all packet transmission and reception, plus the provision of isochronous (same time) data transfer service directly to the application, including the generation of a cycle signal used for timing and synchronization.

The **physical layer (PHY)** translates the logical symbols used by the link layer into electrical signals using differential Data-Strobe (D-S) bit level encoding. It provides the initialization and arbitration services necessary to assure that only one node at a time is sending data (**1394 includes a fair bus mechanism that guarantees all nodes equal access**) and defines the mechanical interface for the serial bus.

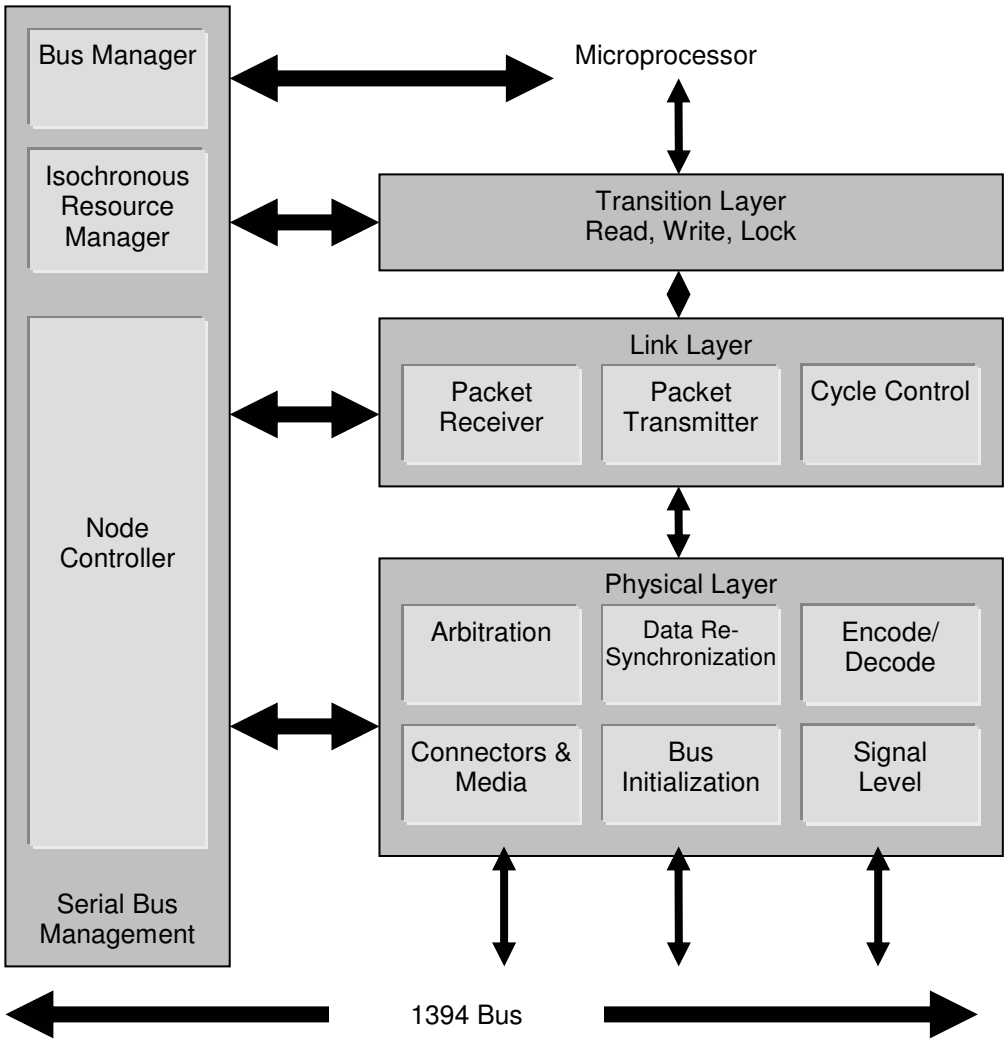

Figure 3 -The 1394 Protocol Stack and Serial bus Management Controller

1394 Bus Management

1394 provides a flexible bus management system that provides the basic control functions and standard control and status registers (CSRs located in the node configuration ROM) needed to control the nodes or to manage bus resources.

- A **cycle master** broadcasts cycle start packets (required for isochronous operation).
- An isochronous **resource manager** centralizes the services needed to allocate bandwidth, channels and other isochronous resources.

- The Bus Manager performs the **power management** procedures and builds a bus **topology map** from every Self ID packet and also produces a **speed map** of the Bus.

Arbitration for cycle master, isochronous resource manager, and bus manager nodes occurs after the bus reset between nodes that are capable of implementing such functions.

Speed Signaling

The 1394 cable standard defines three signaling rates: 98.304, 196.608, and 393.216 Mbps (megabits per second), referred to in the 1394 standard as **S100**, **S200** and **S400**. The signaling rate for the entire bus is ordinarily governed by the slowest active node; however, if a bus manager implements a Topology-Map and a Speed-Map for specific node pairs, the bus can support multiple signaling speeds between individual pairs.

Cables and Connectors

Standard bus interconnections are made with a **6-conductor** cable containing two separately shielded twisted pair transmission lines for signaling, two power conductors, and an overall shield. The **two twisted pairs** are crossed in each cable assembly to create a transmit-receive connection. The **power conductors** (8 to 40 Volts, 1.5 amp max.) supply power to the physical layer for low power devices, isolated devices or any device in repeater mode to keep the physical layer operational when the node power is turned off. The 1394 connector was derived from the childproof Nintendo GameBoy design. Another standard cable/connector defined in the 1394a specification has only 4 conductors. It can be used when the bus power is not required (mainly portable computers and DV cameras).

Isochronous and Asynchronous operation

1394 serial bus data is transmitted using **packet-multiplexing** technique, a packet containing sequences of 32 bits entities called **quadlets** (Fig.4). 1394 supports **asynchronous** and **isochronous** transfer modes. In the asynchronous mode a source and destination nodes are specified in the packets whereas in the isochronous mode, packets are identified by the assigned channel number. The isochronous data transport of the 1394 bus provides the **guaranteed bandwidth and latency** required for high-speed data transfer over multiple channels with on-time delivery.

1394 supports **63** independent isochronous streams (channels) that could carry an unlimited (limited only by the remaining bandwidth) number of audio and video channels. The resource manager includes a BANDWIDTH_AVAILABLE register that specifies the remaining bandwidth available to all nodes with isochronous capability. On bus reset or when an isochronous node is added to the bus, the node requests a bandwidth allocation. As an example, a stereo device requests approximately 3 Mbps of bandwidth at a sampling rate of 48 kHz.

Isochronous transfers are divided in **125 _s isochronous cycles** that are initiated by an asynchronous **Cycle Start packet** sent by the Cycle Master (automatically selected during bus initialization). The isochronous resource manager assigns a channel number (0 to 63) to nodes that request isochronous bandwidth based on values in the manager's CHANNELS_AVAILABLE register. When a node no longer requires isochronous resources, it is expected to release its bandwidth and channel number.

80% of the bus time is reserved for isochronous traffic on the bus, which leaves a minimum of 20% for asynchronous data.

Asynchronous transfers are defined by the IEEE 1212 Control and Status Register (CSR architecture) which describes a protocol for reading, writing and locking memory locations in a node on the bus.

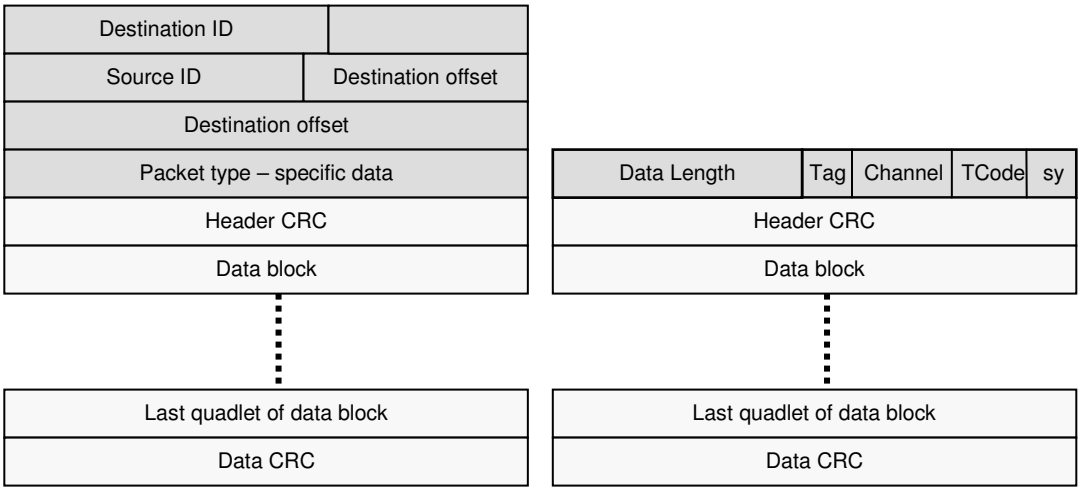

Figure 4- Asynchronous and Isochronous packets.

1394a

The **IEEE Std 1394a-2000** provides some enhancements to 1394 bus performance, which are backward compatible with the original IEEE 1394-1995. The enhancements include the **arbitration accelerations** that improve efficiency of the bus. Also, the **arbitrated short bus reset** allows a node to be added or removed from the bus without causing a long drop in isochronous transmission. Bus reset can be improved from 167 _s (> 125 _s) to 1.3 _s typically. The **PHY pinging** allows the bus manager to calculate the propagation delay between any two-leaf nodes in order to optimize the idle bus time for a specific Bus topology.

Other enhancements:

- Connection debouncing
- Multi-speed packet concatenation
- Token style arbitration
- Software port disconnect

- Incremental bus reconfiguration
- Clarification of ambiguities

1394b

1394b provides further enhancements to 1394 bus performance, which are backward compatible with the original IEEE 1394-1995 in addition it offers:

- More efficient – new arbitration control termed BOSS (Bus Owner Supervisor/Selector) transmits data packets more efficiently, using less network bandwidth.
- New advanced data encoding based on codes used by Gigabit Ethernet & Fiber Channel (called 8B10B) produces less signal distortion.
- Continuous dual simplex transmission
  - one pair transmitting continuously in each direction
  - transmission speed never varies
  - simpler and more efficient than 1394-1995
- GOF, glass optical fiber
- Increased cable length (up to 100 meters vs. 4.5 meters with 1394a)
- Future scalability (future generations already planned at 1.6 Gb/sec & 3.2 Gb/sec
- New CE friendlier 9 pin connector.

Audio Streaming and Control over 1394

The **IEC 61883** International Standard of Digital Interface for Consumer Electronic Audio/Video Equipment defines a scheme for mapping various types of AV data streams onto 1394 isochronous data packets.

Asynchronous data transactions are utilized to send command and control between nodes on the network. IEC 61883 defines **CIP** (Common Isochronous Packet) format for real-time data transfer, **CMP** (Connection Management Procedures) for making isochronous connections between devices and **FCP** (Functional Control Protocol) for exchanging control commands between devices. IEC 61883 also specifies serial bus management capability requirements and the configuration ROM requirements.

The CIP is a standardized method of transferring real-time data streams, such as Digital Video, Audio, and MPEG transport streams, using the isochronous data transfer mechanism of 1394. It defines the CIP header and payload structure, and provides a mechanism to transport timing information.

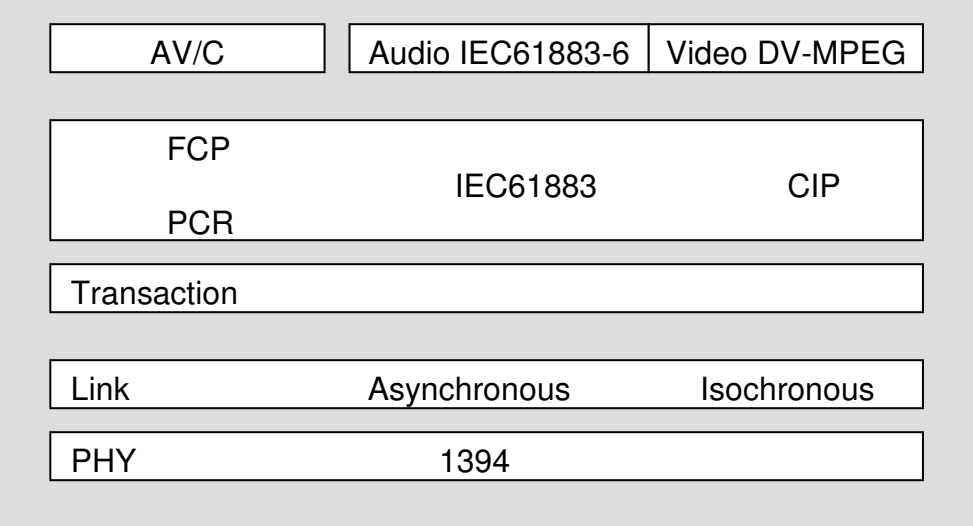

Figure 5 - 1394 stacks for audio

The CMP defines **PCR** (Plug Control Registers) settings sent on the bus through asynchronous transactions. A node that transmits/receives isochronous packets shall have PCRs (Plug Control Registers). The Plug control model enables input and output virtual plugs on every devices receiving or transmitting isochronous data. An output plug on one transmitting node is connected with one or more input plugs on receiving nodes.

The FCP is a standardized method of exchanging control commands. It makes use of command and response frames and supports multiple CTSs (Command Transaction Sets), such as **AV**/**C**.

IEC61883-6 is an extension of IEC61883 and defines the **Audio and Music Data** Transmission protocol. The specification defines audio formats, packetization scheme and sampling clock transmission.

The audio format information and sampling frequency is carried in the IEC61883 CIP. The CIP also carries the data block size and count as well as the **Time Stamp** (SYT) (quantized timing in which an event occurs based on a reference clock, the Cycle_Time). The payload data in a block, if defined as **AM824** (Audio/Music data), contains a label with 24 bit data. The label identifies the data as **Raw Data**, **IEC 958** conformant or **MIDI**.

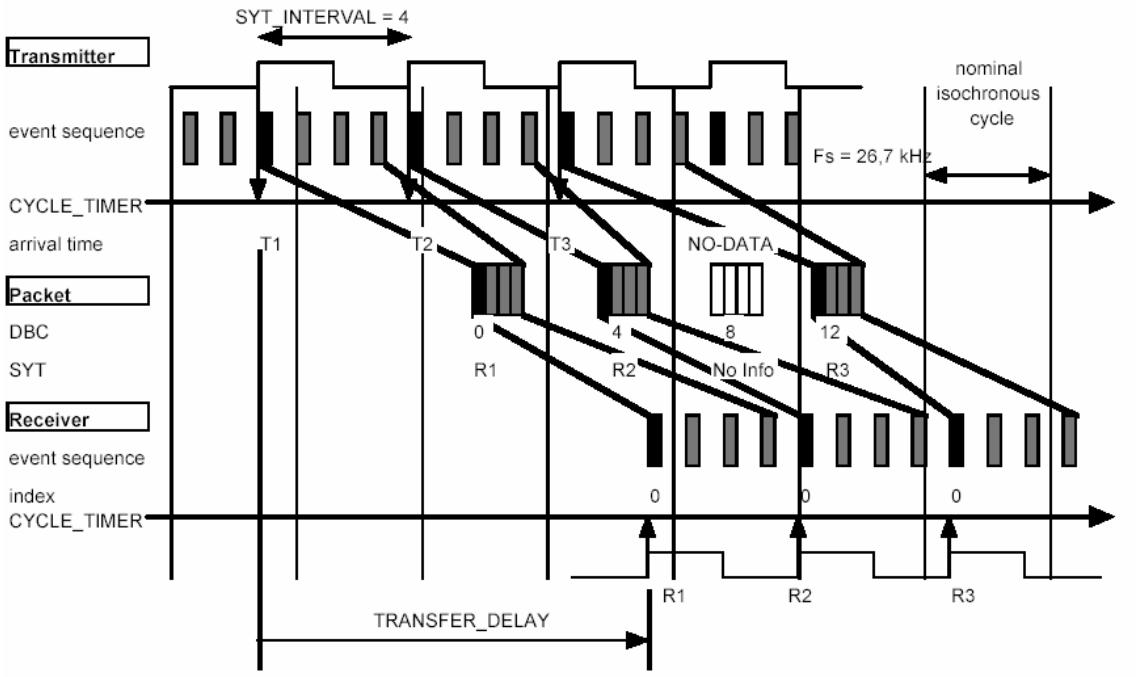

Figure 6 - Blocking and Packetization

The payload data can be transmitted in **Blocking** or **Non-Blocking** modes. In blocking transmission (Fig.6), the transmitter sends a constant number of data in a packet (4 samples) and inserts empty packets or NO-DATA packets when it can't accumulate the required number of events (4). In non-blocking transmission, the transmitter sends a variable number of data, the number of events arrived within the nominal isochronous cycle. The transmitter specifies a presentation time of the event (using **SYT**) at the receiver. The SYT_INTERVAL defines the number of samples between two valid Time Stamps. The TRANSFER_DELAY is added to the Cycle_Time to construct the SYT (Fig.7).

The default **TRANSFER_DELAY** specified in audio and music data specification is **352 µs**, which is the maximum latency time of CIP transmission caused by an arbitrated short bus reset and the insertion of a packet from another isochronous channel transmission. TRANSFER_DELAY can be used for time alignment between streams

(compensate for different audio latency), up to 2 ms delay (the SYT range being from 40.69 ns to 2 ms).

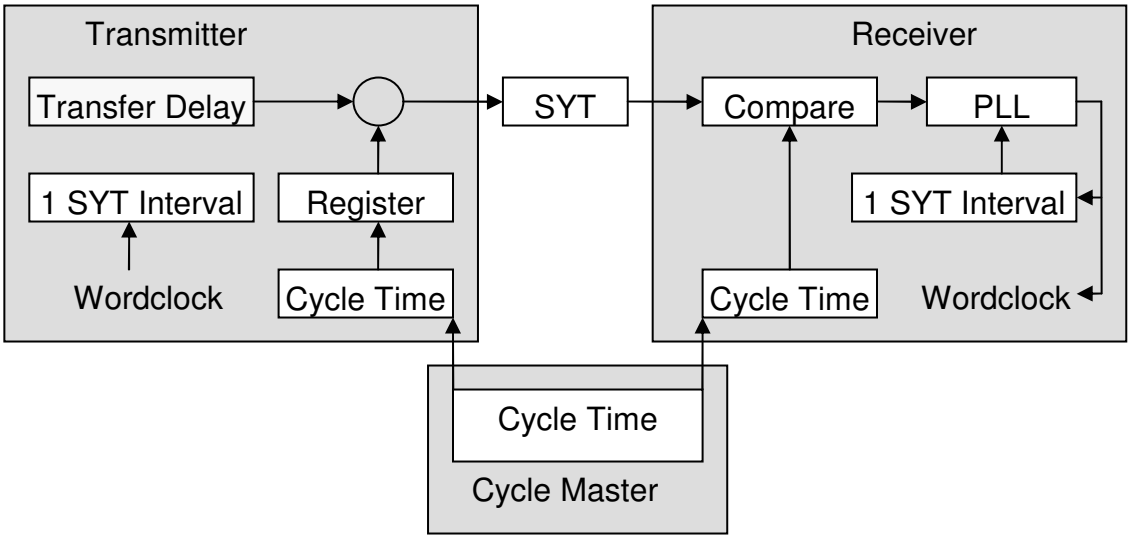

Figure 7 - Time Stamp operation

mLAN

The **mLAN™** specification from Yamaha adds to the **Audio and Music Data** Transmission protocol a Connection Management protocol suitable for electronic musical instruments or professional audio equipment.

Convergence

Computer hardware, software and operating systems already incorporate more and more audio support. For example, **Microsoft WDM Audio** architecture performs all audio processing in kernel mode, which significantly improves latency.

The PC currently processes audio functions such as audio wavetable synthesis, digital audio recording, processing and playback, **MPEG-2** and **Dolby Digital** decoding, sampling rate conversion, compression and decompression algorithms for real time audio over internet (MP3), faster transfer and more efficient storage and 3D positioning (Qsound, Vmax). USB and 1394 provide excellent mechanism for delivering digital audio to external peripherals for high-quality conversion.

PC's are also used in the Recording Studios for sound processing and editing, and special interface cards and external hardware devices are used for conversion to/from analog domain. Most broadcasters are now storing their audio on computer hard disk as data files.

One the home front, Napster began a revolution in music as people downloaded music and created their own CDs and iPOD allowed us to walk around with our own personal music library.

1394 ports are standard on almost all computers sold today and built into both the Windows and Mac operating system.

WDM

WDM (Win32 driver model) architecture has led the way to 1394 support for Windows. It enables bus driver compatibility for Windows 98 though XP and allows dynamically loadable/unloadable drivers, necessary for **1394 Plug & Play**.

The 1394 **OHCI** (Open Host Controller interface) defines standard hardware and software for PC connections to the 1394 bus. It defines register addresses and functions, data structures and DMA models. The OHCI driver is becoming a universal driver for any compliant hardware.

The mLAN audio driver is compliant with the WDM and ASIO architectures. It implements the IEC61883 CIP, CMP, FCP and also supports IEC61883-6 using AM824 data and blocking packetization model. Time Stamp processing can be enabled for direct transmission of 32 to 96 kHz sampled data.

Summary of Advantages

- Open standard. (IEEE-1394)
- High speed, high bandwidth (lots of channels).
- Digital interface (quality, no need to convert - keeps the signal in the digital domain).
- Guaranteed bandwidth and delivery (no dropouts or buffering).
- Single cable for audio and control/data.
- Transmission and reception on the same cable.
- Hot plug-in of nodes and modules. The network instantly identifies and re-configures.
- Parameters can be adjusted, monitored centrally.
- Easy to use - no need for terminators, device IDs or complex set-ups.
- Operates without a computer.
- Defined and accepted audio protocol. (IEC 61883-6)
- Peer to peer (no master/slave - single point of failure).
- Built into Windows' and MAC operating system.
- All Apple/Sony and now most new PC's now come equipped with 1394 ports.

Conclusion
IEEE 1394 offers the audio industry a robust, cost effective, extensible, high speed interconnect. IEEE 1394-enabled PCs play an important role in these applications. IEEE 1394-enabled production studios are a new emerging application that will do for the studio what IEEE 1394 has already done for desktop video editing.

References

[1] IEEE Standard for a High Performance Serial Bus IEEE Std 1394-1995
[2] IEEE Standard for a High Performance Serial Bus IEEE Std 1394a-2000
[3] IEEE Standard for a High Performance Serial Bus IEEE Std 1394b
[4] IEC 61883-1 Consumer audio/video equipment- Digital interface Part 1:General
[5] IEC-PAS 61883-6 Audio and Music Data Transmission Protocol
[6] AV/C Digital Interface Command Set General Specification Version 3.0
[7] AV/C Audio Subunit Specification Version 1.0