

1394 Firewire Hardware Design Considerations

by Burke Henehan

Most multimedia applications share one common denominator: a voracious appetite for I/O bandwidth. How to move massive amounts of data, whether it be graphic, audio, or video, into and out of a computing system or peripheral device can be a daunting task.

The IEEE 1394 Firewire specification defines a next-generation serial I/O subsystem that has the throughput bandwidth for today's multimedia applications as well as expansion capabilities to handle even more demanding applications in the future. Although 1394 can be used as the basis for a backplane-based system with speeds up to 100Mbits/s, this discussion will describe the considerations designers should be aware of when using 1394 as a cable-connected, virtual bus for I/O purposes. The cable version of 1394 currently supports data rates of 100Mbits/s, 200Mbits/s, and 400Mbits/s with higher rates in development.

IEEE 1394 is a transaction-based packet technology that is organized as if it were memory space interconnected between devices. The typical hardware topology of a 1394 I/O network consists of a physical layer (commonly referred to as a PHY) and link layer. The 1394-1995 standard also defines two software layers, a transaction layer and a serial bus management layer, parts of which may be implemented in hardware. The PHY layer has physical signaling circuits and logic that are responsible for power-up initialization, arbitration, bus-reset sensing, and data signaling. The link layer formats data into packets for transmission over a 1394 cable and supports both the asynchronous and isochronous communications modes.

Designing a 1394 videoconferencing camera

Figure 1 shows a block diagram of a typical desktop digital video camera using 1394 for I/O. This article will use such a camera as an illustrative example of how a 1394-based system is designed, but design considerations outside the scope of this fairly simple camera will be included as well so that the discussion will be comprehensive in its scope.

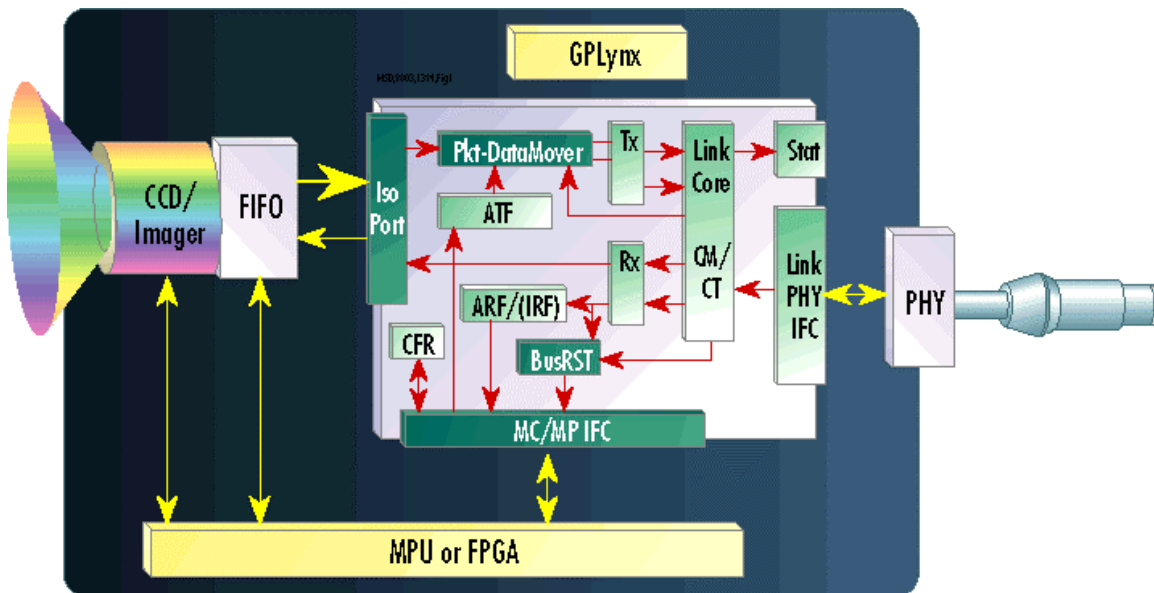


Figure 1

The camera captures images with a charge coupled device (CCD) sensor and stores them in memory. Depending upon the complexity of the application, the camera will use a digital signal processor (DSP) or an application-specific integrated circuit (ASIC) to perform image processing and control functions. The control unit will set up the CCD sensor and initiate the 1394 link layer functions. From the memory, the image is forwarded to the 1394 link layer device where the data is packetized and sent on to the 1394 PHY device for transmission over a 1394 cable. The cable is specified in the 1394-1995 standard, though larger gauge cable can be used if longer distances between devices on the network are required. For the sake of integrated-circuit design simplicity, 1394 PHY and link layer functionality are usually segregated in discrete devices, but this is not always the case and is not required by the specification.

A closer look at the PHY level

The speed of the PHY device used in a design will determine the serial data transfer rate over the 1394 cable. Currently, PHY interface devices support speeds of 100Mbits/s, 200Mbits/s, and 400Mbits/s.

The 1394 Trade Association (TA) has promulgated a specification for 1394-based digital video cameras (current version 1.04). This specification describes the capabilities of a standard 1394-based desktop video conferencing camera, including frame sizes, frame rates, and the packet sizes required for particular modes the camera may use. The six modes and six frame rates defined in the specification cover a wide range. At the lower end is a 160x120 pixel, 24-bits-per-pixel at 7.5 frames per second (FPS) setting that requires a transfer rate somewhat less than 5Mbits/s. Toward the higher end is a 640x480 pixel, 12-bits-per-pixel at 30FPS setting that requires a transfer rate of somewhat more than 120Mbits/s. The highest-quality setting is 640x480 pixels, and 24-bits-per-pixel at 30FPS, which requires a transfer rate of more than 240Mbits/s.

For our example we choose a middle-of-the-road 200Mbits/s PHY, capable of the majority of modes, but leaving out the highest frame rate and highest quality modes.

Cable ports

In a 1394 network topology, a one-port device (one-port PHY) is known as a leaf device because it will be situated at the end of a branch on the network. IEEE 1394 also allows two-ported devices, which can be used to form daisy-chained topologies. Since the 1394 standard mandates a maximum of 16 cable "hops," a maximum of 17 devices can be included in a network, if only one-port and two-port PHY devices are used. Three or more port PHY devices can be formed by using branching as a full 63-node tree topology shown in Figure 2. In such a network, each node acts as a repeater to simulate a single logical bus. The limit to a particular bus segment of 63 nodes is a function of the 1394 address space definition.

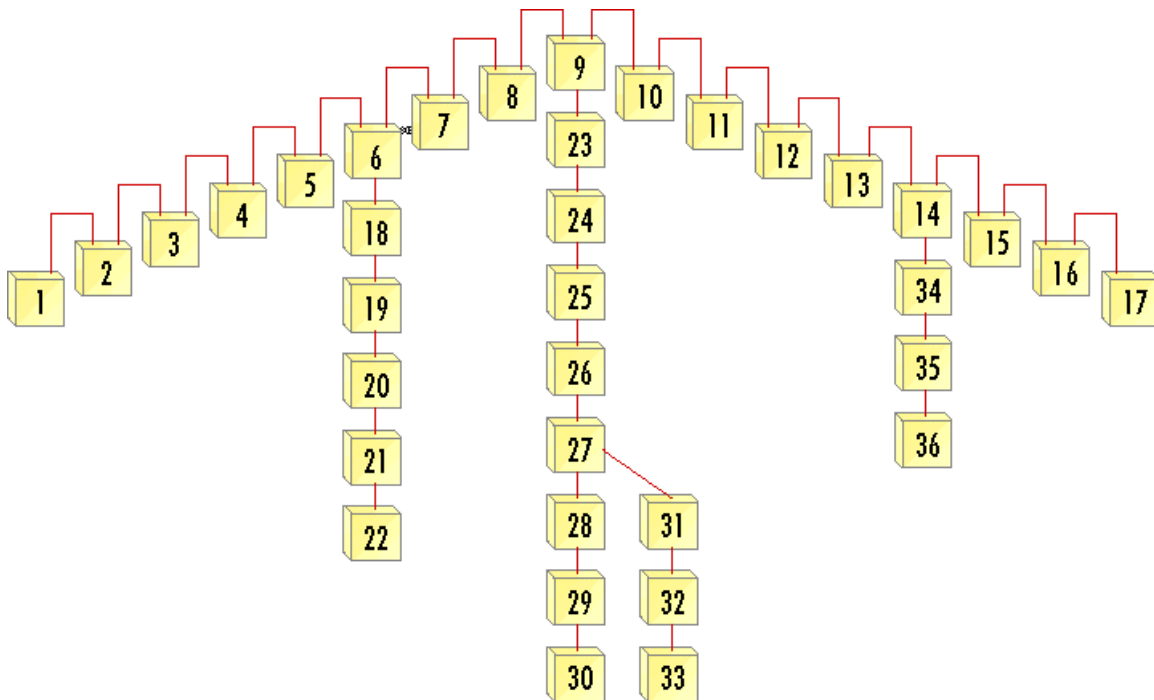


Figure 2

The desktop camera requires only one port since it will only be connected to a PC or similar device. The camera may use the power provided over the 1394 cable as its only power source. This means only a single connection to the outside world is required, since the 1394 cable carries both power and data. When there are no other external connections, the design is simplified by eliminating certain power isolation issues.

Managing power

Power management is an increasing concern in today's systems and several vendors' PHYs include power management capabilities. Most are equipped with three power management pins: power down (PD), link power status (LPS), and cable not active (CNA). Power savings may be achieved when a node's link layer (or link layer interface) is not active. When this occurs, the LPS pin can be asserted to power down the PHY-link interface drivers to save power. When LPS is active, the PHY continues to act as a repeater, maintaining the logical 1394 bus, but the PHY cannot communicate to its link layer until LPS is de-asserted.

The highest level of power saving is achieved when the PD and CNA pins are used. When PD is asserted, the entire PHY is powered down except for the CNA circuitry. The CNA circuitry remains active to monitor the PHY's ports to determine when an active node is plugged into any of its ports. If no ports are detected as connected, the PD pin can be asserted to power down the device. Whenever an active node is plugged in, the CNA circuitry will signal this by asserting the CNA signal and the PHY may be powered up by de-asserting the PD pin. In the case of using the PD pin to deactivate a node, the PHY will stop acting as a repeater on the 1394 bus. Therefore, if the PHY is in the middle of the bus, the bus will break into two separate buses at the powered-down PHY. Since the desktop camera is a leaf node, the last node on a branch, breaking the bus into two segments is not a concern. But power management is a concern, so a power-saving PHY device is an important consideration for a designer.

Other PHY issues

The 1394 signal lines are required to be biased up to a voltage between 1.665V and 2.015V. PHY devices accomplish this by using a twisted-pair (TP) bias pin. On a multiple port PHY with only one TP bias for all ports, if one port fails from a hard short, the other ports on the device will also fail. To prevent this, and allow individual port control, newer PHYs feature a TP bias pin for each port on the device.

The timing for the PHY level must be provided externally from either a crystal or a crystal oscillator. Less expensive crystals are obviously more cost-effective, and 24.576MHz is a standard telecommunications frequency.

Packaging options for PHY interface devices are particularly important in space-sensitive applications such as the digital camera. PHY devices come in several packages, with the lower speeds and smaller number of ports having smaller packages. These range from a 48-pin quad flat pack (QFP) for a one-port, 100Mbits/s PHY; to a 64-pin QFP for a 200Mbits/s, three-port PHY; up to a 100-pin QFP for a 400Mbits/s PHY with six ports.

Link layer considerations

IEEE 1394's link layer defines two communications modes: asynchronous and isochronous. Asynchronous communication packets are guaranteed delivery because when an asynchronous packet is received, the receiver transmits an immediate acknowledgment to the sender. The delivery latency of asynchronous packets cannot be determined and depends upon the traffic on the bus at the time of the communication. Asynchronous packets are targeted to one node on the network or can be sent to all nodes, but can not be broadcast to a subset of nodes on the bus.

IEEE 1394's optional isochronous communications mode guarantees a particular size time slice each $125\mu\text{s}$ (8,000 isochronous cycles per second). Since a device is guaranteed a time slot, and isochronous communication takes precedence over asynchronous, isochronous bandwidth is assured. Ongoing isochronous communication between one or more devices is referred to as a channel. Once a channel has been established, the requesting device is guaranteed to have the requested amount of bus time for that channel every isochronous cycle. Only one device may send data on a particular channel, but targeted broadcast transmissions are possible so that any number of devices may receive data on a channel. A single device may have multiple channels allocated and additional channels may be added as long as isochronous capacity in the device and isochronous bandwidth on the 1394 bus are available.

Certain types of isochronous signals, such as MPEG-2 or digital video (DV) cells, involve particular data transport protocols and formats. When this type of data is sent isochronously over 1394, special packetization techniques are invoked. A common isochronous packet (CIP) header is embedded in the 1394 isochronous packet format. These CIP headers have certain fields, including data length, time stamp, and data block counter, that must be updated or have action taken on them every 125ms. (See Figure 3.) If the link layer device automatically processes CIPs, this task can be offloaded from the system's CPU or control processor.

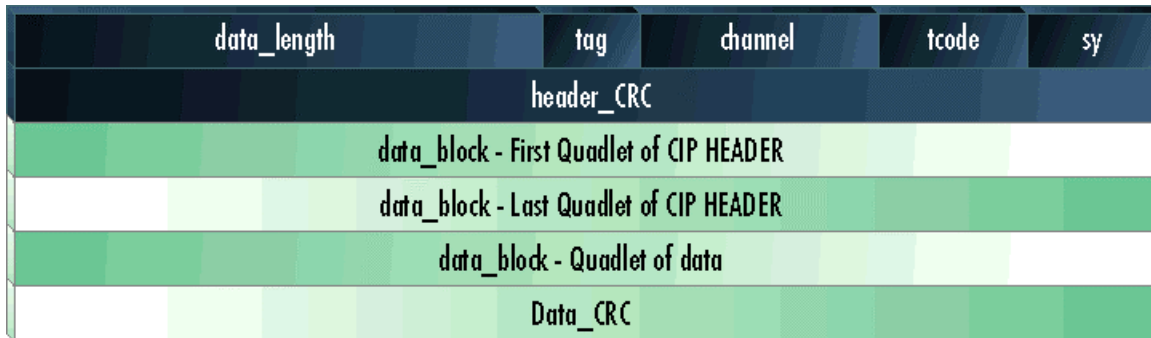


Figure 3

For the purposes of our digital camera design, asynchronous communications won't do since the required amount of 1394 bandwidth cannot be guaranteed. A video camera is essentially a real-time system, requiring dedicated bandwidth to assure the video images will be supplied at a constant rate with no dropouts. As a result, a digital video camera requires both modes of communication. Most of its transmissions will be isochronous data, but it will also require a small amount of asynchronous transmit and receive capability for bus management and device control. For a desktop video camera, 1394 provides enough bandwidth that the video image is sent without any compression or special packetization. Neither the link layer device nor the camera's control processor must handle CIP headers.

Alternative data path configurations

Every application has its own set of idiosyncrasies that affect the characteristics of the system. Depending on the link layer device, various data paths can be established either to the microcontroller or a hardware port. The 1394 link layer device that allows the greatest flexibility in establishing data paths and meets the needs of a particular application should be chosen. For example, a PC host controller may only require one data path, namely from 1394 to the PCI bus. In this case, all 1394 data is funneled to the system's host processor.

Another configuration is to establish two data paths through the link layer (Figure 4). This allows each data path to be customized for the particular data task the application requires. Almost every node requires either a microcontroller or a hardware controller of some sort. However, if the data requires special or high-speed processing that can be done by specialized external hardware, a separate data path can be provided so that the controller need not handle the data. MPEG-2 set-top box chipsets are an example of this. The 1394 link provides a separate path for the MPEG-2 data to connect directly to the MPEG-2 decompression chip. In the design of a digital camera, this two data-path type of configuration would allow for the high-volume bulky isochronous video data to be sent directly onto the 1394 bus without including the microcontroller in the path. The low-volume asynchronous control information can then be sent via a lower bandwidth path to the camera's processor. Since the camera's control processor does not have to handle all data, a less powerful, less expensive control processor may be used. The desktop camera does not use any compression or special packetization so the hardware interface can be simple. The link we choose may use a low power, low cost microcontroller interface and a simple second data path exclusively for isochronous data.

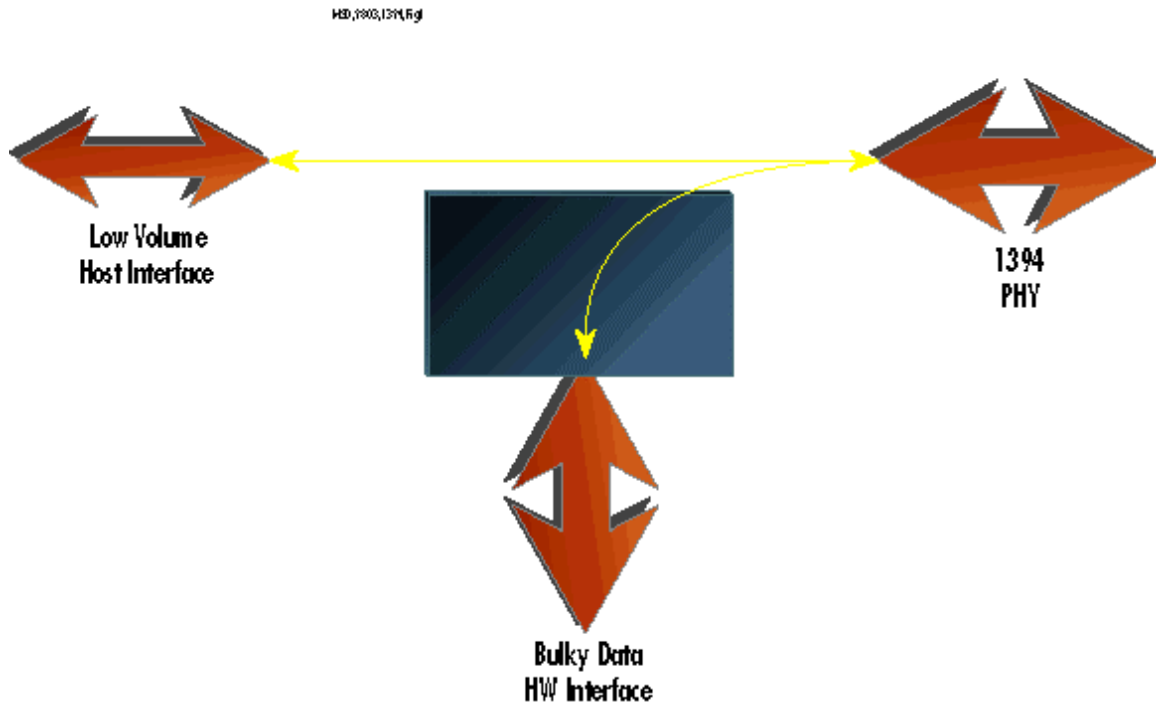


Figure 4

Host processor interfaces

The sort of processor the link layer device will interface with will have an effect on the characteristics of the link. A wide selection of interfaces is available, including generic 32-bit interfaces, PCI bus interfaces, 8/16 bit microcontroller interfaces, and IDE/ATAPI interfaces. For the desktop camera, if it has a high-power 32-bit processor, we may choose a link with a single high-performance 32-bit interface. However, the camera only requires a simple microcontroller and specialized hardware to do the isochronous data transfers. We may choose a link with a simple 8/16-bit interface that also has a separate isochronous data path.

FIFO size vs. DMA

Many of the 1394 link layers available use a store-and-forward scheme of data transfer, meaning the entire packet must be buffered into a FIFO memory before it may be moved out of the FIFO. This is required primarily because these links do not have built-in thresholding direct memory access (DMA) engines. If the link layer device uses store and forward technology, it must have a FIFO large enough to accommodate the maximum packet size used and the latency involved before a packet can be put onto the 1394 bus. If a link layer device supports thresholded DMA, a packet can begin transferring from FIFO memory before the FIFO contains the entire packet. A third method of dealing with this problem is to allow external buffering of large volumes of data. This must be done along with a mechanism to move that data to or from the 1394 bus only when the bus is ready to receive the data or has data to be received by the node. This data mover (DM) capability is a compromise between the extremes of true DMA and very large FIFOs. It also allows a small internal FIFO to accommodate the asynchronous control packets while using a second internally unbuffered data path for high volume isochronous data.

This DM capability is well-suited to the needs of a desktop camera. The camera already has a memory external to the 1394 link layer to hold the images captured by the CCD. The link layer doesn't need full DMA since the image is always in the same memory space and will

always be emptied in the same manner. The interface only needs to be able to start over from the top of the FIFO once a complete image has been transferred.

Endian issues

1394 has been developed as a big-endian architecture that defines the most significant byte as byte zero and the most significant bit as bit zero. If the processor being used is based on a big-endian architecture, there is no problem. However, many processors are based on a little endian architecture that defines the most significant byte as byte 3 (assuming a 32-bit word) and the most significant bit as bit 32. The designer must then accommodate conversions between big-endian and little-endian formats. The problem of converting from big-endian to little-endian and vice versa has been described elsewhere and is outside the scope of this article.

Bus management taxonomy

Nodes on a 1394 bus may vary in complexity and capability as described by five levels. The simplest is transaction-capable, followed by isochronous-capable, cycle master-capable, isochronous resource manager (IRM)-capable, and bus manager-capable.

- If a node is transaction-capable, it is able to respond to asynchronous communication, implement the minimal set of control status registers (CSR), and implement a minimal configuration ROM.
- Isochronous capable nodes add a 24.576MHz clock that increments a cycle timer register that is updated by cycle start packets received over the 1394 bus from the cycle master node. (See IEEE 1394-1995 for a more detailed explanation.)
- Cycle master-capable nodes add the ability to generate the 8kHz cycle start event, generate 1394 cycle start packets, and implement a bus time register.
- IRM-capable nodes perform all the above operations, as well as detect bad self-ID packets, determine the node ID of the chosen IRM, and implement the channels available, bandwidth available, and bus manager ID registers.
- The most complex node function is bus manager (BM). This level adds responsibility for storing every self-ID packet in a topology map and analyzing that topology map to produce a speed map of the entire bus. These two shared maps are used to manage the bus. Finally, the BM must be able to activate the cycle master node, write PHY configuration packets to allow optimization of the bus, and may act as the power manager.

For a 1394 network to perform any isochronous communication, at least one node on the network must be capable of acting as an isochronous resource manager (IRM). In the desktop camera, for instance, we might choose to not implement the IRM function and depend on the PC to act as an IRM. Since an IRM must also be cycle master-capable, the desktop camera need only be isochronous-capable. A PC with a fast processor and lots of memory is an ideal candidate for bus manager responsibilities and operating system vendors are making their systems IRM and bus manager-capable. If we wished the camera to be more capable, we could pick a link that could provide hardware assistance for IRM functions such as self-ID checking or determining the IRM node number to allow a small asynchronous FIFO and a simple host controller.

Enabling multimedia

The demands placed on multimedia systems by application software will only increase in the near future. Users have a voracious appetite for realistic graphics, video and audio, and the market has a way of providing what users want. Multimedia system designers will find that 1394 provides an effective way to move this sort of data into and out of systems. It will be up to the system designer to make the best use of 1394's capabilities and balance these against the other hardware and software capabilities present in the system.

Questions of compatibility

An important compatibility issue involves the PHY and link layer interface. The definition of this interface is not a requirement of IEEE 1394-1995. However, it is critical that the PHY and link layer devices are able to communicate.

An IEEE committee known as 1394, is updating the 1394 specification including standardizing the PHY-link interface. As of this writing (Jan. 1998) this committee is still at work.

Until an industry standard is released and supported by silicon vendors, system designers must verify that the PHY and link layer devices chosen are compatible. This includes verifying several items. Designers should verify compatible setup and hold times across the PHY-link interface. Status transfers from a PHY device to a link layer device are sometimes interrupted. If this occurs, some PHY implementations retransmit the entire status, while others just transmit the bits that had not yet been transferred to the link. During reset, some PHY implementations do not have the system clock (SCLK) available when the PHY reset pin is asserted, but others do.

In addition, if galvanic isolation between the PHY and link is an important issue in a particular design, the method of isolation between the PHY and link devices must be supported by both. Various methods have been used to communicate other functionality between the link and the PHY, including transfers of the link power status (LPS) signal, the configuration manager contender (CMC) signal and the link on (LKON) signal. Designers must verify that the parts chosen are compatible for the functionality desired.