 By Jim Lyle, Silicon Image Inc

EDN®

WHETHER OR NOT YOU BELIEVE IN CONTENT PROTECTION, IF YOU DESIGN DIGITAL-VIDEO PRODUCTS, YOU MUST DEAL WITH THE TECHNOLOGY. HERE'S WHAT YOU NEED TO KNOW TO START APPLYING HIGH-BANDWIDTH DIGITAL-CONTENT PROTECTION.

HDCP: what it is and how to use it

The DVI (digital visual interface) delivers video images with very high resolution and essentially perfect quality. Although this capability is a boon to end users, it has sparked great concern in the entertainment industry, because it raises the specter of unauthorized mass duplication and distribution of "perfect" copies of Hollywood's most valuable content.

Traditional copyright protections (such as infringement lawsuits) are suitable only in specific cases and are impractical on a mass scale. They would be entirely useless against the millions of people who might buy DVDs and copy them for their friends and relatives. Therefore, much of the consumer-electronics equipment available today incorporates copy-protection mechanisms. Different types of devices use different kinds of copy protection. Most techniques stem from cooperation between content providers and equipment manufacturers. For DVI, such a cooperative effort has produced a mechanism called HDCP (high-bandwidth digital-content protection), a two-part cryptographic method to control video delivery.

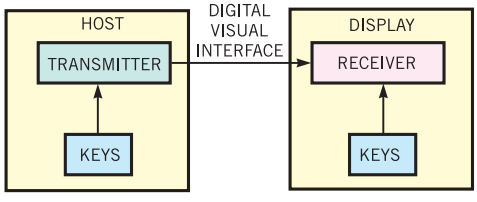
WHAT HDCP IS—AND ISN'T

HDCP is content protection, not copy protection (or more accurately, copy restriction). The difference is subtle but very important. HDCP is not designed to prevent copying or recording, nor will it do away with "time-shifting" (taping of TV programs to pause live video or to enable you to view programs later). The ability to do these things has revolutionized TV viewing. People want these features, and the capabilities will only improve.

Yet, as these features become more capable and ubiquitous, protecting copyrighted material becomes increasingly difficult. The copyrights are important; they protect both those who create materials and those who use them. The Internet raises the stakes because it enables unwitting or unscrupulous low-cost mass distribution of copyrighted material.

Your home-theater system can implement any kind of copy protection it needs. It might allow un-

Figure 1

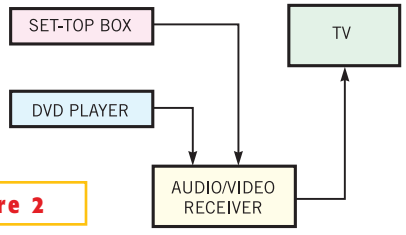


This block diagram represents an HDCP system in its simplest form.

restricted copies, a limited number of copies, limited use of copies, or no copies at all. The exact mechanism depends on the source material, how it is distributed, and the equipment design and configuration. It is important, however, that the designer's decisions on the permissible extent of copying be final.

At this point, HDCP enters the picture. The DVI connection it guards is usually the last link in the video chain. The system's intelligence typically lies somewhere upstream, for example, in the satellite-TV or cable box. Such locations are the right places to decide on copy-protection strategies. HDCP merely protects the choice.

Figure 2



A home-theater system, such as this one, can use an audio/video receiver to route audio and video signals among a variety of inputs and outputs.

SYSTEM ARCHITECTURE

A DVI link is a point-to-point connection with a single transmitter and receiver, so the simplest HDCP system resembles the one that **Figure 1** shows. The host is a PC, a set-top box, a DVD player, or a similar video source that contains the DVI transmitter and a set of HDCP keys. The display is a monitor, flat-panel television, or projector that contains the DVI receiver and a (different) set of HDCP keys.

By design, the HDCP protocol couples a single transmitter to a single receiver. Other devices can't eavesdrop. In some cases, though, the system may need more than one host or display device. One example is a computer that has connections to both a monitor and a projector. Also, many home-theater systems use an AV receiver to route audio and video signals among a variety of inputs and outputs (**Figure 2**).

The HDCP specification defines a repeater function to accommodate configurations such as the one that **Figure 2** shows. A repeater is an active device that has one or more DVI/HDCP inputs and one or more DVI/HDCP outputs. **Figure 3** shows another repeater example. The two outputs and one input represent the PC/monitor/projector example. On one side, the repeater acts as a receiver, accepting video from upstream. On the other side, the repeater acts as a transmitter, sending the video downstream.

Each of these links is separate. Each has a unique transmitter and receiver, and from an HDCP perspective, the data on each is encrypted in a unique way (which is a function of the keys for that particular link). Though separate, the links are not entirely independent: Certain HDCP functions must gather status and other information about the entire system, so the HDCP specification defines how to propagate this information upward through repeaters.

Note that hosts *must* support repeaters. The specification requires this support, and your customers will need the flexibility. A system can also include more than one repeater. Indeed, there may be as many as seven levels of repeaters and 127 receivers. (Repeaters count toward this limit.)

AUTHENTICATION AND REVOCATION

You cannot confidently use HDCP unless both the transmitter and receiver support it and work properly. Authentication tests and verifies these functions and, if unsuccessful, blocks transmission. Authentication must exclude devices that have been compromised or hacked. Revocation accomplishes this exclusion.

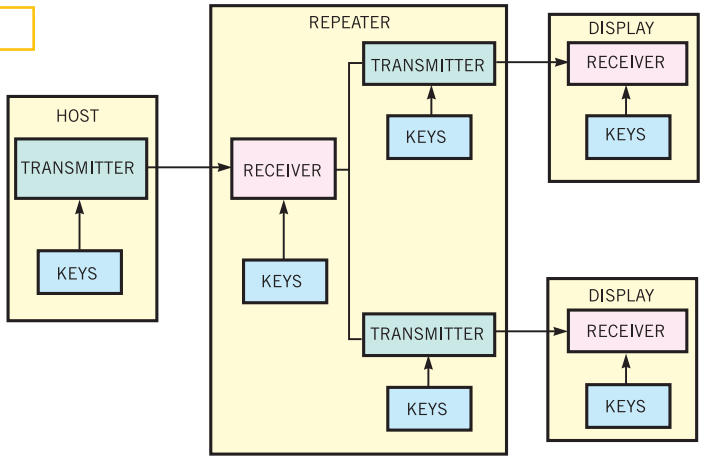
First, transmitters and receivers must demonstrate knowledge of a valid set of keys. The keys themselves are kept private and never revealed, but each side of the link calculates a mathematical result, R_0, which depends on the key values. This calculation also initializes the cipher engines with a secret value, K_S, which forms the video-encryption key.

The transmitter generates a pseudo-random number, A_N, which it sends to the receiver along with the transmitter's KSV (key-selection vector). The receiver then sends its KSV to the transmitter. The KSV values must have the right form—they must contain exactly 20 ones and 20 zeros. If they do, each of the cipher engines independently calculates the R_0 and K_S results.

Finally, the system compares the R_0 values that the transmitter and receiver generate by the transmitter and receiver. The mathematical function prevents the transmitter keys from working in a receiver and the receiver keys from working in a transmitter. Matching R_0 values strongly suggest valid keys.

The flowchart in **Figure 4** shows an example of the firmware operations needed to manage the computation process. This example assumes that the transmitter is Silicon Image's (www.siimage.com) SiI168 device, but any HDCP transmit-

Figure 3



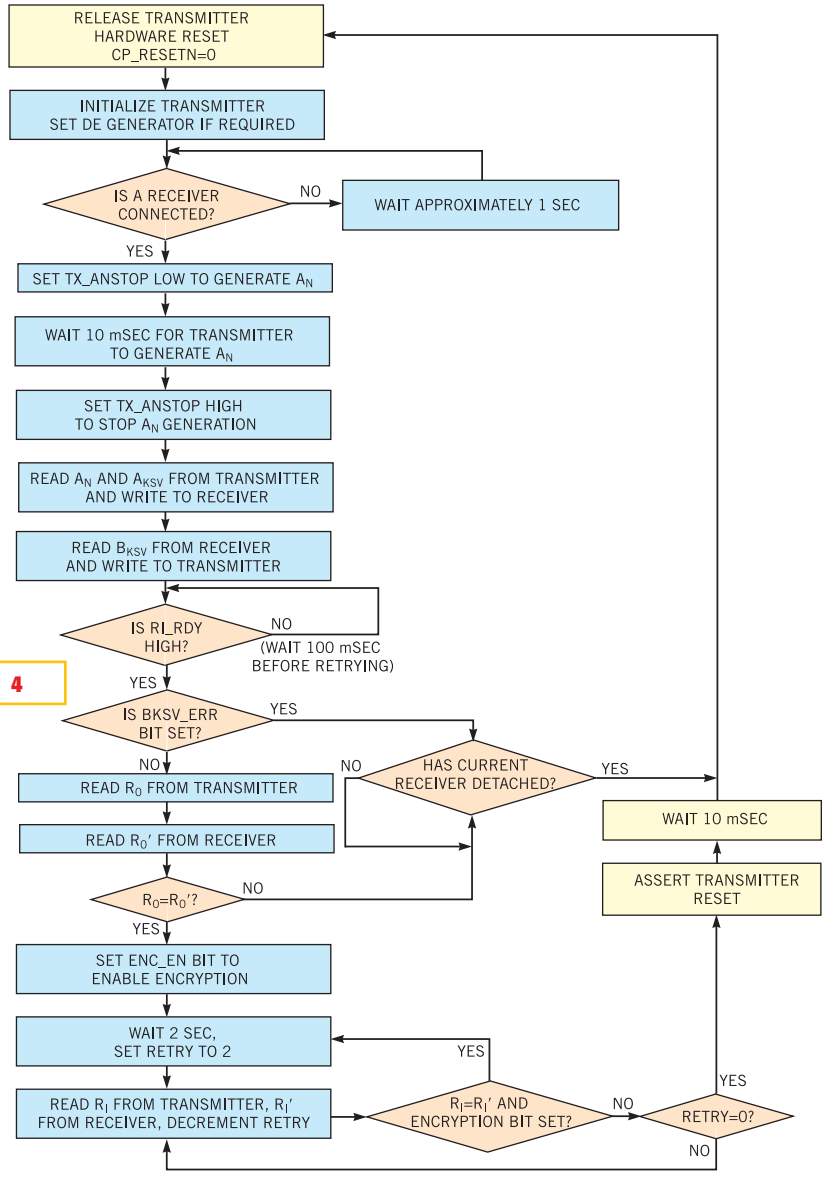
You can use a repeater that has one input and two outputs to drive a monitor and a projector.

ter performs similar functions.

Even valid keys can become compromised (hacked), so HDCP includes a mechanism to revoke keys. The KSV values are unique to each key set and, therefore, to each device. The system can then compare these values to a revocation list, and if either the transmitter or receiver appears on that list, authentication fails. Updates to the revocation list arrive with new media and are automatically integrated. So if a key set somehow does get exposed or copied, the damage can be limited.

This revocation process does not affect other devices, even if the devices are of the same make and model. In that sense, KSV values are like serial numbers. Suppose that Sally and Bob buy the same kind of TV on the same day at the same store. Bob somehow hacks his set, gets caught, and has his KSV value revoked. Sally needn't worry. Her TV has a different KSV value and won't be affected in any way.

Figure 4



This flowchart shows the firmware operations, which manage the information exchange that takes place during authentication.

UPSTREAM AUTHENTICATION

Even if valid hardware and valid keys are present, it might still be possible for an external agent (typically a driver or software application) to interfere. In the worst case, this agent could make it appear that the HDCP hardware is present and active when, in fact, it's not there, not working, or altered in some way. Upstream authentication enables the system to prevent or detect this kind of problem and can also permit the system to move confidential values from the hardware to the software application without fear of the values being observed or altered.

The simplest form of upstream authentication uses purely physical barriers. Many applications have fixed hardware and firmware configurations and are not subject to user-installed upgrades or modifications. The absence of such modifications is typical of consumer-electronics products, including devices such as set-top boxes and DVD players. In such cases, the required security can be established by design, verified by test, and protected by the enclosure and the requisite "no user-serviceable parts" warning label.

A personal computer is completely different, though. Users frequently install their own software and can download new drivers or plug-ins with just a few mouse-clicks. In this environment, upstream authentication combines hardware and software protocols that include encryption and signature algorithms, with the implicit assumption that no driver is trustworthy. Special cryptographic messages that only the software can interpret hide link status and confidential data values. The drivers can only pass the messages along. If the drivers fail to forward the messages or tamper with them in any way, the mischief will be detected.

The original HDCP specification does not define upstream authentication, as it is application-dependent and isn't always necessary. A companion specification defines a recommended implementation for PC applications, however. This specification uses a mathematical system that is similar to HDCP's downstream side. The keys come from a different key space, though, and will not interoperate with

the downstream keys. The specification defines only a limited number of functions. These functions include Status Read (verifies that the HDCP link is working properly), Read M (transfers the confidential M_0 value, which the software application needs), and Read Z (transfers Z_K, which provides initial keying material in some cases).

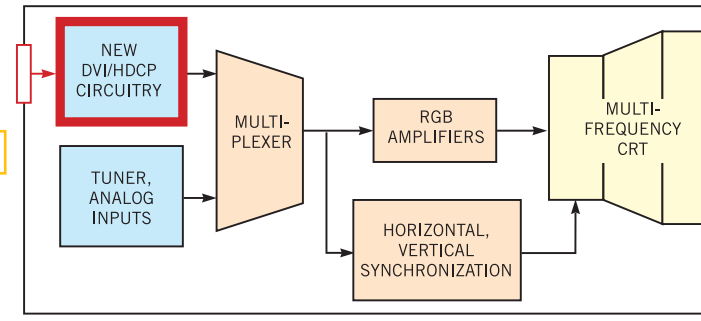


Figure 5

The elements in red represent the items that the manufacturer must add to an analog monitor to equip it with an HDCP-capable DVI.

ENCRYPTION

Encryption, of course, is the brick and mortar of the HDCP strategy, and it prevents eavesdropping. The encryption alters the video data using a reversible function (which XORs the data with the cipher-engine output). Each pixel is separately altered, producing a video image that is thoroughly scrambled and has no recognizable features. If the transmitter and receiver are properly synchronized, the XOR functions cancel, properly reproducing the original video.

During encryption, a link-integrity check occurs approximately every two seconds. Every 128 frames, the receiver and transmitter generate a new value, R_I, which is similar to the original R_0. Comparing these values verifies that the link remains synchronized. This check serves three main purposes. First, it provides additional verification that the initial states were a proper match. Second, it provides ongoing feedback that the link is working properly. Finally, it narrows the window for an outside agent to disrupt the link.

DESIGN CONSIDERATIONS AND PITFALLS

Product designers frequently ask when they will need to be ready for HDCP. The trite-but-true answer—the sooner the better—raises a chicken-and-egg quandary, however. Nobody wants to go to the time, trouble, and cost of building a DVI/HDCP interface into a product if customers can't use it. The market needs both PCs and monitors (or set-top boxes and TVs) before this feature can become effective.

A look at the other side of the coin is important, though. Although having a feature that you don't need isn't good, needing a feature that you don't have can be worse. High-definition, digital-quality, premium video is coming but only to products built with security in mind. Designers who delay implementation risk ignoring their customers' needs.

The good news is that quite a few HDCP-enabled products (both host and display) are either in production or soon will be. Leading manufacturers and content providers have already announced support for DVI/HDCP, and many more will do so in the coming months. So the question is not so much whether you should do this now. It's whether you're already too late.

One special consideration: Television sets are a big investment (some bigger than others), and people tend to hold on to them for a long time. It's doubly important to futureproof these products.

ADDING HDCP TO A PRODUCT

To add HDCP capabilities to your product, you need a few basics. First, you need the DVI: a connector, the DVI silicon, and a small number of discrete components. Then you need the HDCP function, which is generally integrated into the DVI silicon. You also need the HDCP keys and some kind of nonvolatile memory to store them. (The exact kind depends on the HDCP function, but in some cases, the memory also resides in the DVI silicon.)

For example, suppose a monitor or TV manufacturer wants to add an HDCP-capable DVI to an analog product line. **Figure 5** highlights (in red) the necessary new elements. The essential chassis remains the same. One of the analog inputs merely changes to digital. The alterations are isolated, and the manufacturer might even implement them as a daughtercard or optional feature.

The additional circuitry is not complicated; **Figure 6** details one example. The primary elements are the DVI connector (CONN1), the SiI905 HDCP-monitor controller, and an EEPROM. The SiI905 contains all of the DVI and HDCP circuits and produces an RGB analog output. The function is essentially self-contained and requires no external firmware or control. Other varieties of interfaces are also available, including some with digital outputs and some with advanced features, such as scaling and on-screen displays.

Adding an HDCP-capable DVI output to a product such as a graphics adapter or set-top box is a similar exercise. You still need to add the interface silicon, the nonvolatile key memory, and the DVI connector. The circuit complexity is similar to that shown for the receiver. In general, though, the output-side components are not self-contained. HDCP hosts must control the interface and the information exchange. This requirement typically adds software or firmware. The functions aren't complex, but they do add to the development and test effort.

Don't forget that HDCP hosts must also support repeaters. In practice, then, the software or firmware probably has to collect the KSV list and verify its digital signature. HDCP hosts must also solve the upstream-authentication problem, either by explicitly incorporating authentication or, perhaps, by using an architecture that prevents end users from making substantial additions or modifications.

LICENSES, COMPLIANCE, AND ROBUSTNESS

Every manufacturer seeking to build an HDCP product must obtain a license that allows use of the technology and purchase of the requisite keys. In addition, the license sets forth a set of compliance and robustness.

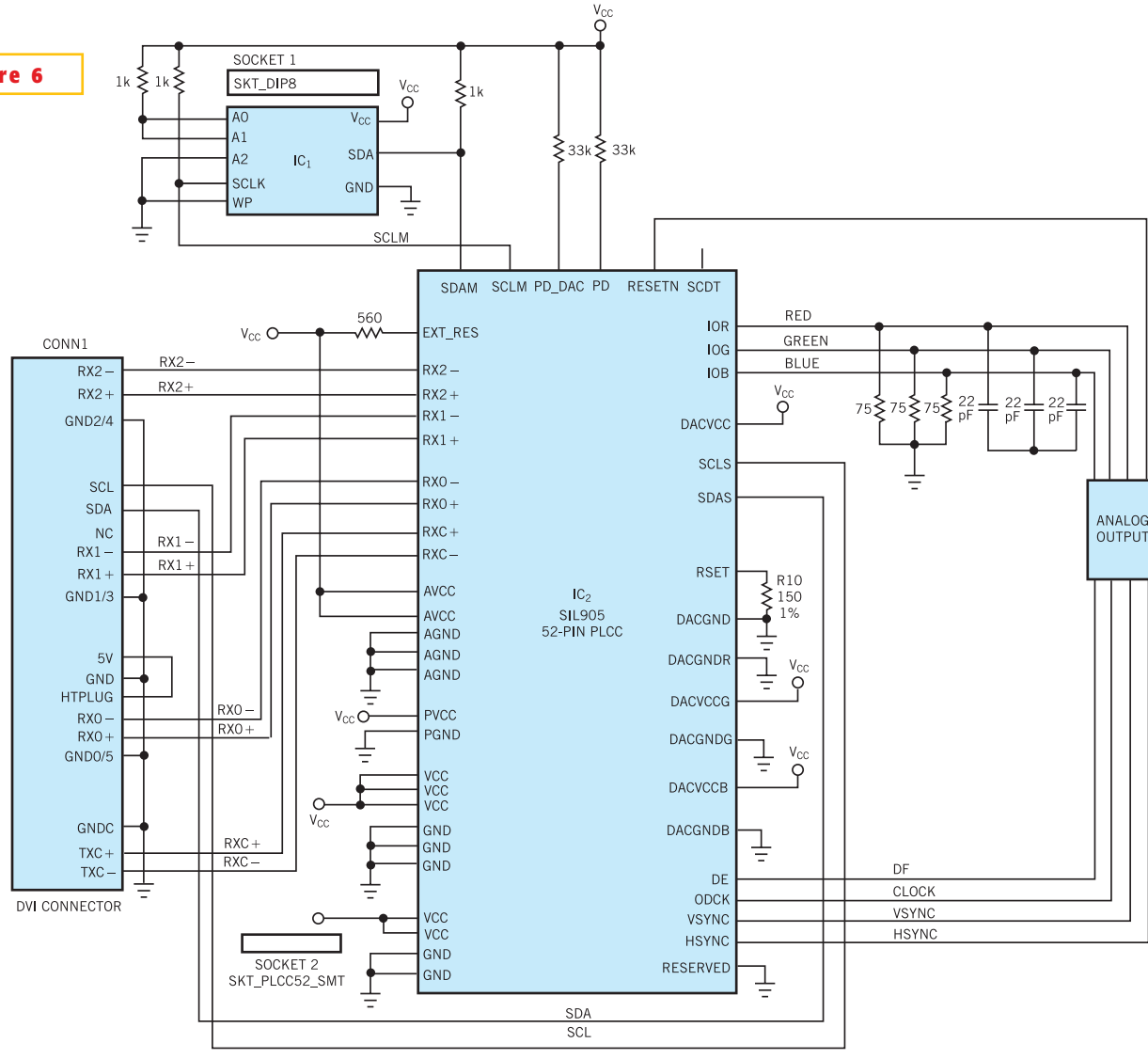
The compliance rules ensure that HDCP products work together, even among different product types and manufacturers. They also limit the permissible types of configurations and place specific restrictions on video copying, buffering, and available outputs. For example, HDCP products may not make copies but are allowed to temporarily buffer video for specified purposes. HDCP products must use unique keys, too, and are not allowed to have decrypted digital or analog outputs.

The robustness rules make it difficult to circumvent the HDCP technology. In other words, an HDCP product must suitably protect its secrets and must not include switches, jumpers, or simple hardware modifications that trivially bypass some of the intended restrictions.

The important point is that, ultimately, each OEM must ensure that its products are both secure and compliant. The interface-silicon manufacturer can help, however, and, indeed, helping is a good way for an IC supplier to add value to its products. In the final analysis, though, equipment manufacturers that integrate all of the HDCP elements must ensure that the final product is compliant and robust.

So what must a product designer do? First, get a copy of the license and study it. (You can find it at the Digital Content Protection, LLC Web site, www.digital-cp.com.) In particular, Exhibit D-1 contains a checklist that asks a series of questions that, in effect, spell out the requirements in exhaustive detail. Second, make sure that your product adequately protects the keys. This protection is critical, and many of the rules are intended to ensure the keys' security. Sometimes, meeting this requirement will require special packaging or encapsulation. Also, pay attention to any manufacturing process that includes the keys. Protecting the keys in the final product is not enough; it's also important to protect them at every stage of design, manufacturing, and test. Find out how the keys are first received and decoded, when and where are they programmed, and whether subcontractors are involved. Find out whether invento-

Figure 6



This schematic of the new circuitry needed for a digital CRT shows that the primary elements are the DVI connector (CONN1), the HDCP-monitor controller, and an EEPROM.

ries need special protection. Find out the answers to all of these questions and more. This requirement need not be daunting, nor must the process be burdensome. Nevertheless, compliance will be easier and cheaper if you think through the issues in advance.

COMPATIBILITY TESTING

You can ensure compatibility among products from different vendors only if all meet some suitable level of specification compliance. Currently, however, no lab or service tests the compliance of HDCP products. Manufacturers need this service, and ongoing discussions aim to establish it. In the meantime, plugfests fill the void.

A plugfest is an increasingly common gathering at which many vendors (including competitors) test their products. Such testing can only indirectly verify a product's compliance, but it does provide good feedback on the product's compatibility. Plugfests are not really conducive to in-depth analysis, because test equipment and test time can be scarce and also because competitors are often hesitant to reveal details or quirks. Still, the events are usually helpful and result in improved interoperability.

For an HDCP product, compatibility testing divides into two equally important parts. The first part concerns the underlying DVI-transport protocol. If the protocol is unreliable or incompatible, no other testing matters. Subtle glitches can cause catastrophic HDCP-link failures, so exhaustive testing of the link's reliability is imperative. Tests must span a wide frequency range and use different voltages and jitter levels.

HDCP testing includes both authentication and encryption phases. Plugfest testing usually uses the public keys published in the HDCP specification. If possible, repeat this testing at different resolutions and frequencies and with different cables and configurations. Let the test run for a while; glitches that can break the link or confuse the logic may not occur for several minutes. Try unusual operations, too. Change the channel, reauthenticate while the link is active, and do anything else you can think of to expose system quirks.

If you are testing a host device that serves as the link master, you need to provide the test software that drives the entire authentication process. Make the software flexible. Include the ability to override values, such as A_n; change device addresses and modes; and dump and alter registers. If at all possible, bring the source code, a compiler, and the program's author.

If you are testing a display device, you can rely on the other guy to provide the link-control software. Receivers are slaves that use standardized register maps. However, if the receiver incorporates firmware, all of the previous comments about the host software apply. Come prepared to change the software and to examine variables and registers.

The real challenge at these events is to find a way to be as helpful as possible within the constraints of the competitive environment. A thorough test-and-debug session sometimes requires disclosure of details that one side really doesn't want to share with the other. Such sessions also involve risk, because design flaws can surface while competitors are watching. Still, the benefits are enormous if you can get past these concerns and concentrate on the test at hand. Take copious notes, especially of unexplained behaviors and phenomena. Follow up on them too, either at the event or back at the office.

The best time to find bugs, of course, is before they get designed in. Toward this end, carefully read the specification. Don't forget the errata or the license, either, because they contain additional clarifications and restrictions.

FUTURE DIRECTIONS

As it is currently defined, DVI 1.0 is specified solely for connection between a computer and a monitor. (The present DVI license contains language to that effect.) That situation is changing, however, as more consumer-electronics devices incorporate DVIs.

DVI and HDCP are evolving in subtle ways to meet the needs of this new marketplace. You can see the changes in smaller connectors, longer and lower cost cables, audio support, and alternate color spaces. The licenses and the specifications will be updated accordingly, and the result may have a new name to differentiate it from the existing DVI and HDCP interfaces. Backward compatibility is a top priority, though, so that today's designs don't become obsolete tomorrow.

The DVI 1.0 interface provides premium-quality performance. With the addition of HDCP technology, it constitutes the preferred means of delivering copyrighted high-bandwidth material. If you're a manufacturer, you should start thinking now about adding DVI to your products. The design is not difficult, and it will ensure that your products can ride the coming wave of high-definition content and services.□

AUTHOR'S BIOGRAPHY

Jim Lyle is a senior staff engineer at Silicon Image, where he has focused on the system architecture and integration of the company's DVI and HDCP products. He holds a degree in electrical engineering from the University of California at Berkeley and has worked at Tandem Computers, Sun Microsystems, and National Semiconductor. His other interests include antique radios and Scottish country dancing.

Reprinted from EDN, April 18, 2002 by Reprint Services. 223300