

# **HyperTransport™ Technology I/O Link**

**A High-Bandwidth I/O Architecture**

**ADVANCED MICRO DEVICES, INC.**

One AMD Place  
Sunnyvale, CA 94088

**(#25012A)**

***Abstract***

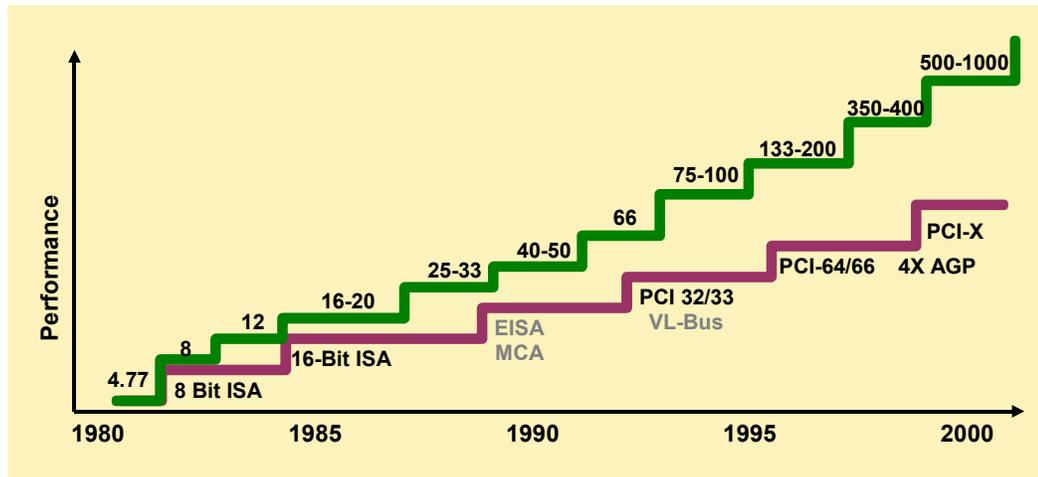
This white paper describes AMD’s HyperTransport™ technology, a new I/O architecture for personal computers, workstations, servers, high-performance networking and communications systems, and embedded applications. This scalable architecture can provide significantly increased bandwidth over existing bus architectures and can simplify in-the-box connectivity by replacing legacy buses and bridges. The programming model used in HyperTransport technology is compatible with existing models and requires little or no changes to existing operating system and driver software.

***Contents***

The I/O Bandwidth Problem..... 3  
 The HyperTransport™ Technology Solution..... 4  
     Original Design Goals..... 5  
     Flexible I/O Architecture ..... 6  
     Device Configurations ..... 7  
 Technical Overview..... 9  
     Physical Layer ..... 9  
         Minimal Pin Count..... 9  
         Enhanced Low-Voltage Differential Signaling..... 11  
         Greatly Increased Bandwidth ..... 12  
     Data Link Layer ..... 13  
         Initialization ..... 13  
     Protocol and Transaction Layers..... 13  
         Commands..... 14  
         Data Packets..... 15  
         Address Mapping ..... 15  
         I/O Stream Identification ..... 15  
         Ordering Rules ..... 16  
     Session Layer..... 17  
         Standard Plug ‘n Play Conventions ..... 17  
         Minimal Device Driver Porting..... 18  
         Link Width Optimization ..... 18  
         Link Frequency Initialization..... 19  
 Implementation Examples..... 19  
     Daisy Chain ..... 19  
     Switched Environment..... 20  
     Multiprocessor System ..... 20  
 HyperTransport™ Technology Specifications..... 21  
 HyperTransport™ Technology Consortium ..... 22  
 Industry Support for HyperTransport™ Technology ..... 23  
 Summary ..... 24  
 Glossary ..... 24  
 AMD Overview ..... 25  
 Cautionary Statement ..... 25

## The I/O Bandwidth Problem

While microprocessor performance continues to double every eighteen months, the performance of the I/O bus architecture has lagged, doubling in performance approximately every three years, as illustrated in Figure 1.



**Figure 1. Trends in I/O Bus Performance**

This I/O bottleneck constrains system performance, resulting in diminished actual performance gains as the processor and memory subsystems evolve. Over the past 20 years, a number of legacy buses, such as ISA, VL-Bus, AGP, LPC, PCI-32/33, and PCI-X, have emerged that must be bridged together to support a varying array of devices. Servers and workstations require multiple high-speed buses, including PCI-64/66, AGP Pro, and SNA buses like InfiniBand. The hodge-podge of buses increases system complexity, adds many transistors devoted to bus arbitration and bridge logic, while delivering less than optimal performance.

A number of new technologies are responsible for the increasing demand for additional bandwidth.

- ❑ High-resolution, texture-mapped 3D graphics and high-definition streaming video are escalating bandwidth needs between CPUs and graphics processors.
- ❑ Technologies like high-speed networking (Gigabit Ethernet, InfiniBand, etc.) and wireless communications (Bluetooth) are allowing more devices to exchange growing amounts of data at rapidly increasing speeds.
- ❑ Software technologies are evolving, resulting in breakthrough methods of utilizing

multiple system processors. As processor speeds rise, so will the need for very fast, high-volume inter-processor data traffic.

While these new technologies quickly exceed the capabilities of today's PCI bus, existing interface functions like MP3 audio, v.90 modems, USB, 1394, and 10/100 Ethernet are left to compete for the remaining bandwidth. These functions are now commonly integrated into core logic products.

Higher integration is increasing the number of pins needed to bring these multiple buses into and out of the chip packages. Nearly all of these existing buses are single-ended, requiring additional power and ground pins to provide sufficient current return paths. High pin counts increase RF radiation, which makes it difficult for system designers to meet FCC and VDE requirements. Reducing pin count helps system designers to reduce power consumption and meet thermal requirements.

In response to these problems, AMD began developing the HyperTransport™ I/O link architecture in 1997. HyperTransport technology has been designed to provide system architects with significantly more bandwidth, low-latency responses, lower pin counts, compatibility with legacy PC buses, extensibility to new SNA buses, and transparency to operating system software, with little impact on peripheral drivers.

## The HyperTransport™ Technology Solution

HyperTransport technology, formerly codenamed Lightning Data Transfer (LDT), was developed at AMD with the help of industry partners to provide a high-speed, high-performance, point-to-point link for interconnecting integrated circuits on a board. With a top signaling rate of 1.6 GHz on each wire pair, a HyperTransport technology link can support a peak aggregate bandwidth of 12.8 Gbytes/s.

The HyperTransport I/O link is a complementary technology for InfiniBand and 1Gb/10Gb Ethernet solutions. Both InfiniBand and high-speed Ethernet interfaces are high-performance networking protocol and box-to-box solutions, while HyperTransport is intended to support “in-the-box” connectivity.

The HyperTransport specification provides both link- and system-level power management capabilities optimized for processors and other system devices. The ACPI-compliant power management scheme is primarily message-based, reducing pin-count requirements.

HyperTransport technology is targeted at networking, telecommunications, computer and high performance embedded applications and any other application in which high speed, low latency, and scalability is necessary.

The general features and functions of HyperTransport technology are summarized in Table 1.

**Table 1. Feature and Function Summary**

<b>Feature/Function</b>	<b>HyperTransport Technology</b>
<i>Bus Type</i>	Dual, unidirectional, point-to-point links
<i>Link Width</i>	2, 4, 8, 16, or 32 bits
<i>Protocol</i>	Packet-based, with all packets multiples of four bytes (32 bits). Packet types include Request, Response, and Broadcast, any of which can include commands, addresses, or data.
<i>Bandwidth (Each Direction)</i>	100 to 6400 Mbytes/s
<i>Data Signaling Speeds</i>	400 MHz to 1.6 GHz
<i>Operating Frequencies</i>	400, 600, 800, 1000, 1200, and 1600 Megatransfers/second
<i>Duplex</i>	Full
<i>Max Packet Payload or Burst Length</i>	64-byte packet
<i>Power Management</i>	ACPI-compatible
<i>Signaling</i>	1.2-V Low-Voltage Differential Signaling (LVDS) with a 100-ohm differential impedance
<i>Multiprocessing Support</i>	Yes
<i>Environment</i>	Inside the box
<i>Memory model</i>	Coherent and noncoherent

### Original Design Goals

In developing HyperTransport technology, the architects of the technology considered the design goals presented in this section. They wanted to develop a new I/O protocol for “in-the-box” I/O connectivity that would:

- ❑ **Improve system performance**
  - Provide increased I/O bandwidth
  - Reduce data bottlenecks by moving slower devices out of critical information paths
  - Reduce the number of buses within the system
  - Ensure low latency responses
  - Reduce power consumption
- ❑ **Simplify system design**
  - Use a common protocol for “in-chassis” connections to I/O and processors
  - Use as few pins as possible to allow smaller packages and to reduce cost

- **Increase I/O flexibility**
  - Provide a modular bridge architecture
  - Allow for differing upstream and downstream bandwidth requirements
- **Maintain compatibility with legacy systems**
  - Complement standard external buses
  - Have little or no impact on existing operating systems and drivers
- **Ensure extensibility to new system network architecture (SNA) buses**
- **Provide highly scalable multiprocessing systems**

### **Flexible I/O Architecture**

The resulting protocol defines a high-performance and scalable interconnect between CPU, memory, and I/O devices. Conceptually, the architecture of the HyperTransport I/O link can be mapped into five different layers, which structure is similar to the Open System Interconnection (OSI) reference model.

In HyperTransport technology:

- The *physical layer* defines the physical and electrical characteristics of the protocol. This layer interfaces to the physical world and includes data, control, and clock lines.
- The *data link layer* includes the initialization and configuration sequence, periodic cyclic redundancy check (CRC), disconnect/reconnect sequence, information packets for flow control and error management, and doubleword framing for other packets.
- The *protocol layer* includes the commands, the virtual channels in which they run, and the ordering rules that govern their flow.
- The *transaction layer* uses the elements provided by the protocol layer to perform actions, such as reads and writes.
- The *session layer* includes rules for negotiating power management state changes, as well as interrupt and system management activities.

These functions are completely described in the HyperTransport technology specifications, and several are discussed briefly in this white paper.

## Device Configurations

HyperTransport technology creates a packet-based link implemented on two independent, unidirectional sets of signals. It provides a broad range of system topologies built with three generic device types:

- *Cave*—A single-link device at the end of the chain.
- *Tunnel*—A dual-link device that is not a bridge.
- *Bridge*—Has a primary link upstream link in the direction of the host and one or more secondary links.

Example configurations include:

- A cave device connected directly to a host bridge.
- A chain of tunnel devices connected to a host bridge.
- Multiple chains of tunnel devices connected to a bridge, which is then connected to a host bridge.
- Multiple chains of tunnel devices connected to a switch, which is then connected to a host bridge.
- Any combination of the above.

Figure 2 shows several examples of the different device configurations possible in HyperTransport technology. In these figures, “P” indicates a primary interface and “S” indicates a secondary interface. (A brief glossary of some terminology used to describe the HyperTransport protocol can be found on page 24.) More detailed block diagrams for some example implementations are shown in the section beginning on page 19.

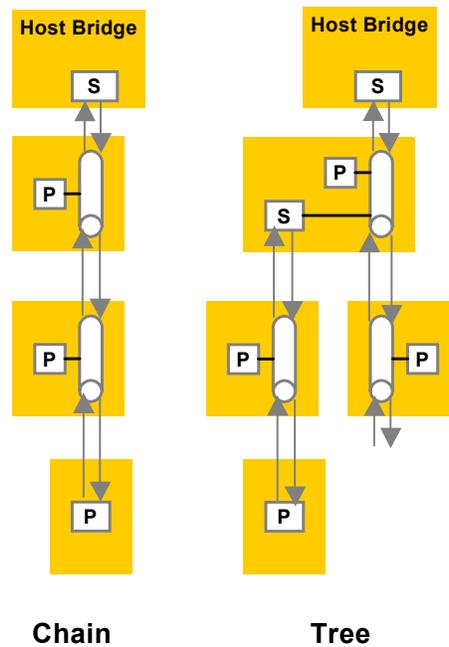
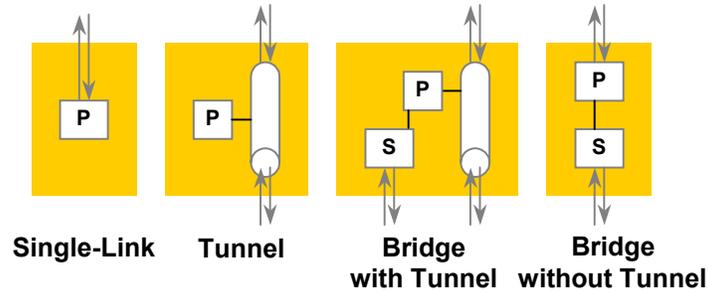


Figure 2. Example HyperTransport™ Technology Device Configurations



- The signals used in HyperTransport technology are summarized in Table 2.
- Commands, addresses, and data (CAD) all share the same bits.
  - Each data path includes a Control (CTL) signal and one or more Clock (CLK) signals.
    - The CTL signal differentiates commands and addresses from data packets.
    - For every grouping of eight bits or less within the data path, there is a forwarded CLK signal. Clock forwarding reduces clock skew between the reference clock signal and the signals traveling on the link. Multiple forwarded clocks limit the number of signals that must be routed closely in wider HyperTransport links.
  - For most signals, there are two pins per bit.
  - In addition to CAD, Clock, Control,  $V_{LDT}$  power, and ground pins, each HyperTransport device has Power OK (PWROK) and Reset (RESET#) pins. These pins are single-ended because of their low-frequency use.
  - Devices that implement HyperTransport technology for use in lower power applications such as notebook computers should also implement Stop (LDTSTOP#) and Request (LDTREQ#). These power management signals are used to enter and exit low-power states.

**Table 2. Signals Used in HyperTransport™ Technology**

Signal Name	Description	Comment
CAD	<b>Commands, Addresses and Data:</b> Carries command, address, or data information.	CAD width can be different in each direction.
CTL	<b>Control:</b> Used to distinguish control packets from data packets.	
CLK	<b>Clock:</b> Forwarded clock signal.	Each byte of CAD has a separate clock signal. Data is transferred on each clock edge.
PWROK	<b>Power OK:</b> Power and clocks are stable.	Single-ended.
RESET#	<b>HyperTransport Technology Reset:</b> Resets the chain.	Single-ended.
LDTSTOP#	<b>HyperTransport Technology Stop:</b> Enables and disables links during system state transitions.	Used in systems requiring power management. Single-ended.
LDTREQ#	<b>HyperTransport Technology Request:</b> Requests re-enabling links for normal operation.	Used in systems requiring power management. Single-ended.

**Enhanced Low-Voltage Differential Signaling**

The signaling technology used in HyperTransport technology is a type of low voltage differential signaling (LVDS). However, it is not the conventional IEEE LVDS standard. It is an enhanced LVDS technique developed to evolve with the performance of future process technologies. This is designed to help ensure that the HyperTransport technology standard has a long lifespan. LVDS has been widely used in these types of applications because it requires fewer pins and wires. This is also designed to reduce cost and power requirements because the transceivers are built into the controller chips.

HyperTransport technology uses low-voltage differential signaling with differential impedance ( $Z_{OD}$ ) of 100 ohms for CAD, Clock, and Control signals, as illustrated in Figure 4. Characteristic line impedance is 60 ohms. The driver supply voltage is 1.2 volts, instead of the conventional 2.5 volts for standard LVDS. Differential signaling and the chosen impedance provide a robust signaling system for use on low-cost printed circuit boards. Common four-layer PCB materials with specified di-electric, trace, and space dimensions and tolerances or controlled impedance boards are sufficient to implement a HyperTransport I/O link. The differential signaling permits trace lengths up to 24 inches for 800 Mbit/s operation.

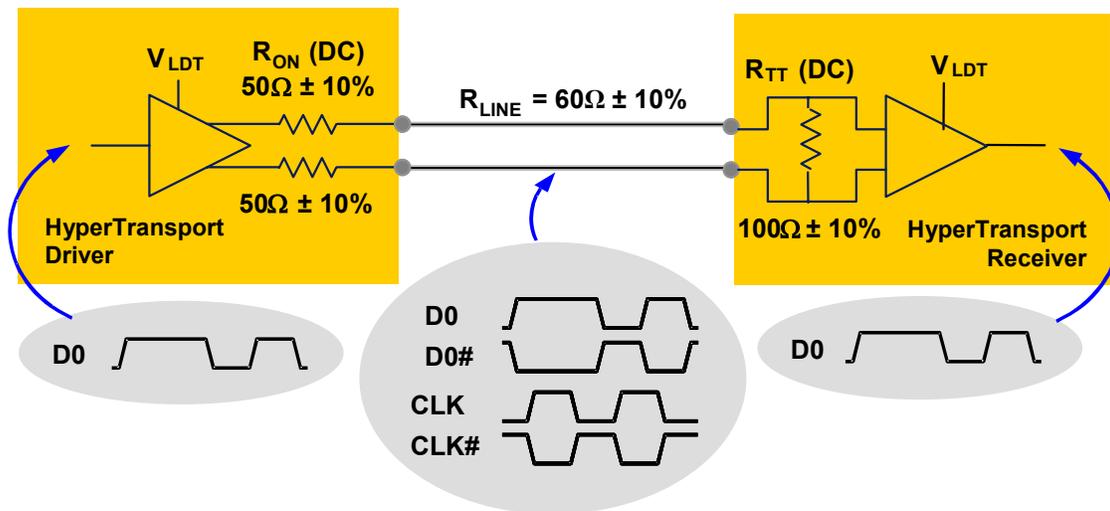


Figure 4. Enhanced Low-Voltage Differential Signaling (LVDS)

At first glance, the signaling used to implement a HyperTransport I/O link would seem to increase pin counts because it requires two pins per bit and uses separate upstream and downstream data paths. However, the increase in signal pins is offset by two factors:

- By using separate data paths, HyperTransport I/O links are designed to operate at much higher frequencies than existing bus architectures. This means that buses delivering equivalent or better bandwidth can be implemented using fewer signals.
- Differential signaling provides a return current path for each signal, greatly reducing the number of power and ground pins required in each package.

**Greatly Increased Bandwidth**

Commands, addresses, and data traveling on a HyperTransport link are *double-pumped*, where transfers take place on both the rising and falling edges of the clock signal. For example, if the link clock is 800 MHz, the data rate is 1600 MHz.

- An implementation of HyperTransport links with 16 CAD bits in each direction with a 1.6-GHz data rate provides bandwidth of 3.2 Gigabytes per second in each direction, for an aggregate peak bandwidth of 6.4 Gbytes/s, or 48 times the peak bandwidth of a 33-MHz PCI bus.
- A low-cost, low-power HyperTransport link using two CAD bits in each direction and clocked at 400 MHz provides 200 Mbytes/s of bandwidth in each direction, or nearly four times the peak bandwidth of PCI 32/33.

Such a link can be implemented with just 24 pins, including power and ground pins, as shown in Table 3.

**Table 3. Total Pins Used for Each Link Width**

Link Width (Each Way)	2	4	8	16	32
<i>Data Pins (total)</i>	8	16	32	64	128
<i>Clock Pins (total)</i>	4	4	4	8	16
<i>Control Pins (total)</i>	4	4	4	4	4
<b>Subtotal (High Speed)</b>	<b>16</b>	<b>24</b>	<b>40</b>	<b>76</b>	<b>148</b>
<i>V<sub>LDT</sub></i>	2	2	3	6	10
<i>GND</i>	4	6	10	19	37
<i>PWROK</i>	1	1	1	1	1
<i>RESET#</i>	1	1	1	1	1
<b>Total Pins</b>	<b>24</b>	<b>34</b>	<b>55</b>	<b>103</b>	<b>197</b>

## Data Link Layer

The data link layer includes the initialization and configuration sequence, periodic cyclic redundancy check (CRC), disconnect/reconnect sequence, information packets for flow control and error management, and doubleword framing for other packets. These topics are discussed in detail in the *HyperTransport™ Technology Input/Output Link Protocol Specification* (#23888). This section of the white paper includes a brief discussion of the initialization process supported by HyperTransport technology.

### *Initialization*

HyperTransport technology-enabled devices with transmitter and receiver links of equal width can be easily and directly connected. Devices with asymmetric data paths can also be linked together easily. Extra receiver pins are tied to logic 0, while extra transmitter pins are left open. During power-up, when RESET# is asserted and the Control signal is at logic 0, each device transmits a bit pattern indicating the width of its receiver. Logic within each device determines the maximum safe width for its transmitter. While this may be narrower than the optimal width, it provides reliable communications between devices until configuration software can optimize the link to the widest common width.

For applications that typically send the bulk of the data in one direction, component vendors can save costs by implementing a wide path for the majority of the traffic and a narrow path in the lesser used direction. Devices are not required to implement equal-width upstream and downstream links.

## Protocol and Transaction Layers

The protocol layer includes the commands, the virtual channels in which they run, and the ordering rules that govern their flow. The transaction layer uses the elements provided by the protocol layer to perform actions, such as read request and responses. These topics are discussed in more detail in the *HyperTransport™ Technology Input/Output Link Protocol Specification* (#23888).

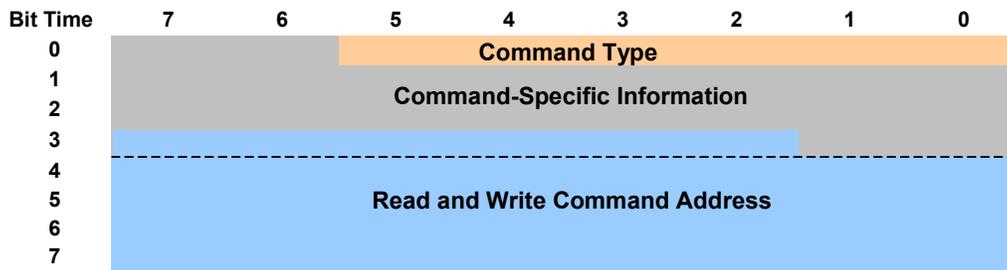
**Commands**

All HyperTransport technology commands are either four or eight bytes long and begin with a 6-bit command type field. The most commonly used commands are Read Request, Read Response, and Write. The basic commands are summarized in Table 4, listed by *virtual channel*. A virtual channel contains requests or responses with the same ordering priority.

**Table 4. Basic HyperTransport™ Technology Commands**

Virtual Channel	Command	Comment
<i>Posted</i>	Posted Write	Followed by data packet(s).
	Broadcast	Issued by host bridge downstream to communicate information to all devices.
	Fence	All posted requests in a stream cannot pass it.
<i>Non-Posted</i>	Non-Posted Write	
	Read	Designates whether response can pass posted requests or not.
	Flush	Forces all posted requests to complete.
	Atomic Read-Modify-Write	Generated by I/O devices or bridges and directed to system memory controlled by the host.
<i>Responses</i>	Read Response	Response to read command, is followed by data packet(s).
	Target Done	A transaction not requiring returned data has completed at its target.

Figure 5 shows the command format.

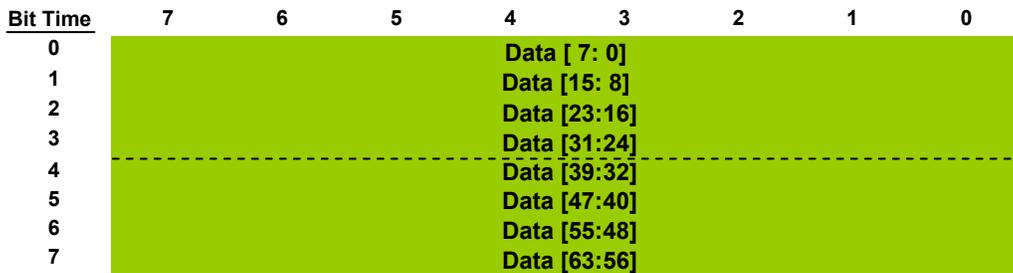


**Figure 5. Command Format**

When the command requires an address, the last byte of the command is concatenated with an additional four bytes to create a 40-bit address.

**Data Packets**

A Write command or a Read Response command is followed by data packets. Data packets are four to 64 bytes long in four-byte increments. Figure 6 shows a packet of eight bytes.



**Figure 6. Eight-Byte Data Packet**

Transfers of less than four bytes are padded to the four-byte minimum. Byte-granularity reads and writes are supported with a four-byte mask field preceding the data. This is useful when transferring data to or from graphics frame buffers where the application should only affect certain bytes that may correspond to one primary color or other characteristics of the displayed pixels. A control bit in the command indicates whether the writes are byte or doubleword granularity.

**Address Mapping**

Reads and writes to PCI I/O space are mapped into a separate address range, eliminating the need for separate memory and I/O control lines or control bits in read and write commands.

Additional address ranges are used for in-band signaling of interrupts and system management messages. A device signaling an interrupt performs a byte-granularity write command targeted at the reserved address space. The host bridge is responsible for delivery of the interrupt to the internal target.

**I/O Stream Identification**

Communications between the HyperTransport host bridge and other HyperTransport technology-enabled devices use the concept of *streams*. A HyperTransport link can handle multiple streams between devices simultaneously. HyperTransport technology devices are *daisy-chained*, so that some streams may be passed through one node to the next.

Packets are identified as belonging to a stream by the Unit ID field in the packet header, as shown in Figure 7. There can be up to 32 unique IDs within a HyperTransport chain. Nodes within a HyperTransport chain may contain multiple units.

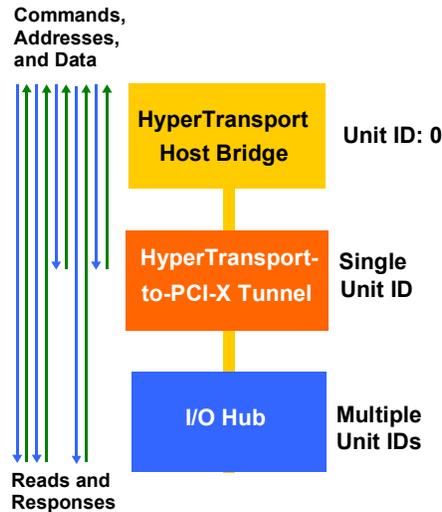


Figure 7. I/O Streams Use Unit IDs

It is the responsibility of each node to determine if information sent to it is targeted at a device within it. If not, the information is passed through to the next node. If a device is located at the end of the chain and it is not the target device, an error response is passed back to the host bridge.

Commands and responses sent from the host bridge have a Unit ID of zero. Commands and responses sent from other HyperTransport technology devices on the chain have their own unique ID.

If a bus-mastering HyperTransport technology device like a RAID controller sends a write command to memory above the host bridge, the command will be sent with the Unit ID of the RAID controller. HyperTransport technology permits posted write operations so that these devices do not wait for an acknowledgement before proceeding. This is useful for large data transfers that will be buffered at the receiving end.

**Ordering Rules**

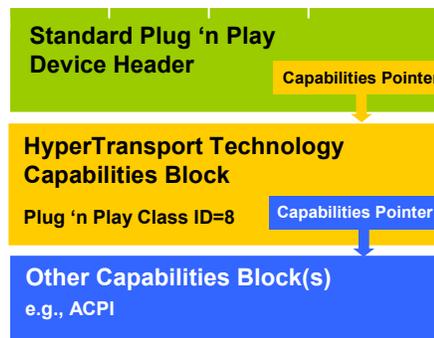
Within streams, the HyperTransport I/O link protocol implements the same basic ordering rules as PCI. Additionally, there are features that allow these ordering rules to be relaxed. A Fence command aligns posted cycles in all streams, and a Flush command flushes the posted write channel in one stream. These features are helpful in handling protocols for bridges to other buses such as PCI, InfiniBand, AGP.

**Session Layer**

The session layer includes link width optimization and link frequency optimization along with interrupt and power state capabilities. These topics are discussed in more detail in the *HyperTransport™ Technology Input/Output Link Protocol Specification* (#23888).

**Standard Plug ‘n Play Conventions**

Devices enabled with HyperTransport technology use standard “Plug ‘n Play” conventions for exposing the control registers that enable configuration routines to optimize the width of each data path. AMD registered the HyperTransport Specific Capabilities Block with the PCI SIG. This Capabilities Block, illustrated in Figure 8, permits devices enabled with HyperTransport technology to be configured by any operating system that supports a PCI architecture.

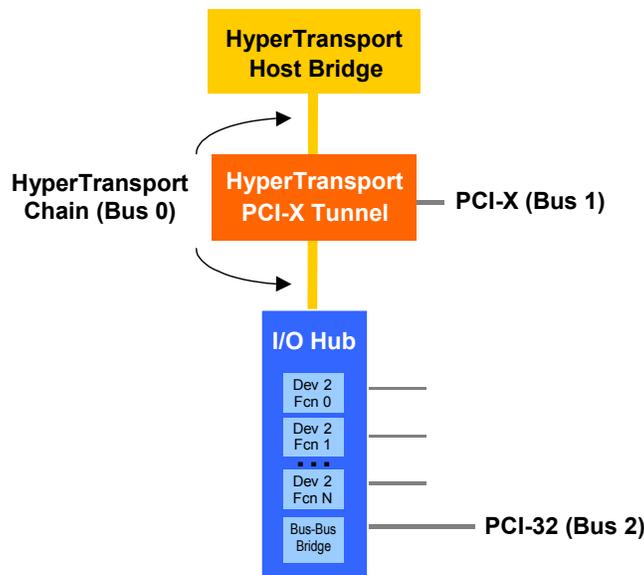


**Figure 8. HyperTransport™ Technology Capabilities Block**

Since system enumeration and power-up are implementation-specific, it is assumed that system firmware will recognize the Capabilities Block and use the information within it to configure all HyperTransport host bridges in the system. Once the host bridges are identified, devices enabled with HyperTransport technology that are connected to the bridges can be enumerated just as they are for PCI devices. Configuration information that is collected and the structures created by this process will look to a Plug ‘n Play-aware operating system (OS) just like those of PCI devices. In short, the Plug ‘n Play-aware OS does not require any modification to recognize and configure devices enabled with HyperTransport technology.

***Minimal Device Driver Porting***

Drivers for devices enabled with HyperTransport technology are unique to the devices just as they are to PCI I/O devices, but the similarities are great. Companies that build a PCI I/O device and then create an equivalent device enabled with HyperTransport technology should have no problems porting the driver. To make porting easier, the chain from a host bridge is enumerated like a PCI bus, and devices and functions within a device enabled with HyperTransport technology are enumerated like PCI devices and functions, as shown in Figure 9.



**Figure 9. Familiar Bus and Device Enumeration**

***Link Width Optimization***

The initial link-width negotiation sequence may result in links that do not operate at their maximum width potential. All 16-bit, 32-bit, and asymmetrically-sized configurations must be enabled by a software initialization step. At cold reset, all links power-up and synchronize according to the protocol. Firmware (or BIOS) then interrogates all the links in the system, reprograms them to the desired width, and takes the system through a warm reset to change the link widths.

Devices that implement the LDTSTOP# signal can disconnect and reconnect rather than enter warm reset to invoke link width changes.

### *Link Frequency Initialization*

At cold reset, all links power-up with 200-MHz clocks. For each link, firmware reads a specific register of each device to determine the supported clock frequencies. The reported frequency capability, combined with system-specific information about the board layout and power requirements, is used to determine the frequency to be used for each link. Firmware then writes the two frequency registers to set the frequency for each link. Once all devices have been configured, firmware initiates an LDTSTOP# disconnect or RESET# of the affected chain to cause the new frequency to take effect.

## Implementation Examples

### **Daisy Chain**

HyperTransport technology has a daisy-chain topology, giving the opportunity to connect multiple HyperTransport input/output bridges to a single channel. HyperTransport technology is designed to support up to 32 devices per channel and can mix and match components with different link widths and speeds.

This capability makes it possible to create HyperTransport technology devices that are building blocks capable of spanning a range of platforms and market segments. For example, a low-cost entry in a mainstream PC product line might be designed with an AMD Duron™ processor. With very little redesign work, as shown in Figure 10, this PC design could be upgraded to a high-end workstation by substituting high-end AMD Athlon™ processors and bridges with HyperTransport technology to expand the platform's I/O capabilities.

Figure 10 also illustrates the concept of tunnels, in which multiple HyperTransport tunnels can be daisy-chained onto a single I/O link. A tunnel can be viewed as a basic building block for complex system designs.

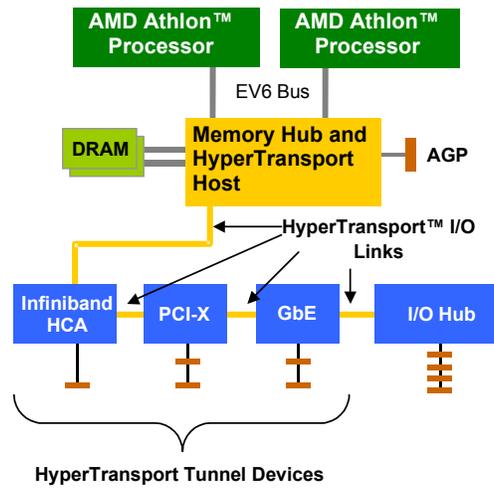


Figure 10. Example Daisy Chain Implementation

**Switched Environment**

A number of industry partners are developing HyperTransport switches, allowing engineers to have a great deal of flexibility in their system designs. In this type of configuration, a HyperTransport I/O switch handles multiple HyperTransport I/O data streams and manages the interconnection between the attached HyperTransport devices. For example, a four-port HyperTransport switch could aggregate data from multiple downstream ports into a single high-speed uplink, or it could route port-to-port connections. A switched environment allows multiple high-speed data paths to be linked while simultaneously supporting slower speed buses.

**Multiprocessor System**

AMD is using a superset of HyperTransport technology to handle cache coherency in its multiprocessor server systems. In the model of a four-processor server shown in Figure 11, each of the Northbridges has two coherent HyperTransport links to create an array of four CPUs. Each core logic chipset has its own memory and a HyperTransport link available for I/O expansion.

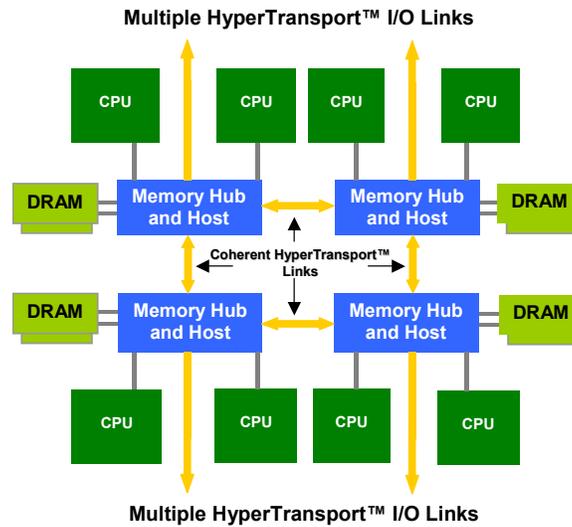


Figure 11. Example Multiprocessor Implementation

## HyperTransport™ Technology Specifications

The following HyperTransport technology specifications and design guides are available.

- ❑ *HyperTransport™ Technology Input/Output Link Protocol Specification (#23888)* defines the HyperTransport technology signals, packets, commands, interrupts, configuration accesses, address map, error handling, clocking, and initialization.
- ❑ *HyperTransport™ Technology Electrical Specification (#23890)* defines the electrical characteristics of the HyperTransport link. It includes characteristics for power supplies and power dissipation, DC and AC output signals including test load circuits, DC and AC input signals, impedance requirements, link transfer timing, and phase recovery timing.
- ❑ *HyperTransport™ Technology Uniprocessor Protocol Specification (#24684)* describes the coherent form of the HyperTransport protocol, which can be used to connect a processor to an external Northbridge device.
- ❑ *HyperTransport™ Technology Interface Design Guide (#24734)* includes guidelines for designing signal interconnect and power distribution, including the device, device package, and PCB interconnect.
- ❑ *HyperTransport™ Technology PHY Interface Specification (#24735)* specifies the behavior and signaling of a commonly-used physical interface implementation.

In general, the primary specifications of interest to most audiences will be the *HyperTransport™ Technology Input/Output Link Protocol Specification* (#23888) and the *HyperTransport™ Technology Electrical Specification* (#23890).

Motherboard designers, chip-set designers, and embedded system designers should read the *HyperTransport™ Technology Interface Design Guide* (#24734) for physical implementation details and guidelines for die, package, and board design.

Chip designers who are interested in seeing the interface that AMD uses between its core logic and a HyperTransport technology PHY should read the *HyperTransport™ Technology PHY Interface Design Guide* (#24735).

Firmware developers and BIOS software developers should read the *HyperTransport™ Technology Firmware Design Guide* (#25016).

For more information about the specifications, call: 800-538-8450, ext. 44775

Or visit this website: <http://www.hypertransport.org>

## HyperTransport™ Technology Consortium

AMD is releasing the specifications to an industry-supported non-profit trade association in the fall of 2001.

The HyperTransport Consortium will manage and refine the specifications, and promote the adoption and deployment of HyperTransport technology. It is also expected to consist initially of a Technical Working Group and a Marketing Working Group. Subordinate task forces will do the work of the consortium. Anticipated technical task forces include:

- ❑ Protocol Task Force
- ❑ Connectivity Task Force
- ❑ Graphics Task Force
- ❑ Technology Task Force
- ❑ Power Management Task Force

Information on joining the HyperTransport Technology Consortium can be found at this website: <http://www.hypertransport.org>

## **Industry Support for HyperTransport™ Technology**

AMD has disclosed HyperTransport technology specifications under non-disclosure agreement (NDA) to over 170 companies interested in building products that incorporate this technology.

Multiple partners have signed the license agreement for HyperTransport technology, including, among many others:

Sun Microsystems	Cisco Systems	Broadcom
Texas Instruments	NVIDIA	Acer Labs
Hewlett-Packard	Schlumberger	Stargen
PLX Technology	Mellanox	FuturePlus
API Networks	Altera	LSI Logic
PMC-Sierra	Pericom	Transmeta

AMD has worked with Teradyne and Schlumberger as vendors of equipment to test HyperTransport technology implementations. Several products are already available for this purpose from Teradyne.

The first products enabled with HyperTransport technology are planned to be in volume production by late 2001, including API's HyperTransport-to-PCI I/O bridge and NVIDIA's new nForce chipset platform for PCs. In addition, Altera and Xilinx have announced high-performance, high-density programmable logic devices (PLDs) that will be designed into networking, communication and Internet devices.

HyperTransport technology is also finding a place in widely diversified processor architectures, including x86, Sparc, MIPS, and embedded CPUs. The adoption of HyperTransport technology in so many different CPU architectures is a strong testament to its inherent value in increasing I/O bandwidth.

Many HyperTransport devices are under development for each of these segments:

Desktop PCs	Workstations
Mobile PCs	Embedded Systems
Servers	Consumer Electronics
LAN Routers and Switches	

## Summary

HyperTransport technology is a new high-speed, high-performance, point-to-point link for integrated circuits. It provides a universal connection designed to reduce the number of buses within the system, provide a high-performance link for embedded applications, and enable highly scalable multiprocessing systems. It is designed to enable the chips inside of PCs and networking and communications devices to communicate with each other up to 48 times faster than with existing technologies. HyperTransport technology provides an extremely fast connection that complements externally visible bus standards like the PCI, as well as emerging technologies like InfiniBand and Gigabit Ethernet. HyperTransport technology is truly the universal solution for in-the-box connectivity.

## Glossary

For reference, the following terms are used in the HyperTransport protocol:

- *Bit-time*—Half of a clock period in duration. Two data bits are transmitted on each signal per cycle.
- *Bridge*—A HyperTransport technology bridge device has a primary link, that being the upstream link in the direction of the host, and one or more secondary links.
- *Cave*—A single-link device at the end of a chain.
- *Chain*—A single HyperTransport chain contains no HyperTransport-to-HyperTransport bridge devices. It may contain native HyperTransport technology peripheral devices (like an Ethernet controller) and may also contain bridges to other interconnects (like PCI). A HyperTransport chain is terminated at one or both ends by a bridge. In the simplest topology, a HyperTransport chain connects to the host bridge at one end and has no connection at the other end.
- *Fabric*—A HyperTransport I/O fabric is implemented as one or more daisy chains of HyperTransport technology devices, with a bridge to the host system at one end.
- *Host bridge*—The interface from the host to the HyperTransport chain.
- *Host*—Can contain multiple bridges, each supporting either a single HyperTransport chain or a tree of HyperTransport chains.
- *Node*—A physical entity that connects to one end of a HyperTransport link.
- *Packet*—A series of bit-times. Forms the basis of communication between two nodes.

- *Source*—The node that initiates a transaction.
- *Stream*—An I/O stream is a collection of transactions that can be treated independently in the fabric with respect to ordering rules. A given I/O stream always originates from the same node.
- *Target*—The node that ultimately services the transaction on behalf of the source. Note that there may be intermediary nodes between the source and the target.
- *Transaction*—A sequence of packets that are exchanged between two or more nodes in the system and that result in a transfer of information.
- *Tunnel*—A dual-link device that is not a bridge. Tunnel devices act as I/O building blocks.

## AMD Overview

AMD is a global supplier of integrated circuits for the personal and networked computer and communications markets with manufacturing facilities in the United States, Europe, and Asia. AMD produces microprocessors, flash memory devices, and support circuitry for communications and networking applications. Founded in 1969 and based in Sunnyvale, California, AMD had revenues of \$4.6 billion in 2000. (NYSE: AMD).

## Cautionary Statement

This white paper includes forward-looking statements that are made pursuant to the safe harbor provisions of the Private Securities Litigation Reform Act of 1995. Forward-looking statements are generally preceded by words such as “expects,” “plans,” “believes,” “anticipates,” or “intends.” Investors are cautioned that all forward-looking statements in this white paper involve risks and uncertainties that could cause actual results to differ from current expectations. Forward-looking statements in this white paper about the HyperTransport technology involve the risk that HyperTransport technology will not function as designed; that AMD may not be successful in developing an infrastructure to support the HyperTransport technology; that third parties may not provide infrastructure solutions to support the HyperTransport technology; and that software applications will not be optimized for use with the HyperTransport technology. We urge investors to review in detail the risk and uncertainties in the company’s Securities and Exchange Commission filings, including the most recently filed Form-10K.

© 2001 Advanced Micro Devices, Inc.

AMD, the AMD logo, AMD Athlon, AMD Duron and combinations thereof, and HyperTransport are trademarks of Advanced Micro Devices, Inc. Other product names used in this publication are for identification purposes only and may be trademarks of their respective companies.