The need for Hypertransport

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.
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
Hypertransport design goals

**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
Features of HyperTransport

- Its point to point approach.
- Only two device per connection.
- Low voltage differential signaling on high speed paths.
- Precise control over PCB trace length & routing.
- Source synchronous clock using Dual Data Rate (DDR) technology.
- Sophisticated protocol eliminate retries, disconnect & wait states.
- Provides support for isochronous traffic.
- CRC error detection and elimination.
- Support for buses for legacy support.
- PCI compatibility for minimal impact on OS and driver software.
- Provides Networking support.
**Greatly Increased Bandwidth**

Commands, addresses, and data traveling on a HyperTransport link are *doublepumped*, 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>
<thead>
<tr>
<th>Link Width (Each Way)</th>
<th>2</th>
<th>4</th>
<th>8</th>
<th>16</th>
<th>32</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Pins (total)</td>
<td>8</td>
<td>16</td>
<td>32</td>
<td>64</td>
<td>128</td>
</tr>
<tr>
<td>Clock Pins (total)</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>8</td>
<td>16</td>
</tr>
<tr>
<td>Control Pins (total)</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td><strong>Subtotal (High Speed)</strong></td>
<td>16</td>
<td>24</td>
<td>40</td>
<td>76</td>
<td>148</td>
</tr>
<tr>
<td>$V_{LDT}$</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>6</td>
<td>10</td>
</tr>
<tr>
<td>GND</td>
<td>4</td>
<td>6</td>
<td>10</td>
<td>19</td>
<td>37</td>
</tr>
<tr>
<td>PWROK</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>RESET#</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td><strong>Total Pins</strong></td>
<td>24</td>
<td>34</td>
<td>55</td>
<td>103</td>
<td>197</td>
</tr>
<tr>
<td>Feature/Function</td>
<td>HyperTransport Technology</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>----------------------------------</td>
<td>--------------------------------------------------------------------------------------------</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Bus Type</strong></td>
<td>Dual, unidirectional, point-to-point links</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Link Width</strong></td>
<td>2, 4, 8, 16, or 32 bits</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Protocol</strong></td>
<td>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.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Bandwidth (Each Direction)</strong></td>
<td>100 to 6400 Mbytes/s</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Data Signaling Speeds</strong></td>
<td>400 MHz to 1.6 GHz</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Operating Frequencies</strong></td>
<td>400, 600, 800, 1000, 1200, and 1600 Megatransfers/second</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Duplex</strong></td>
<td>Full</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Max Packet Payload or Burst Length</strong></td>
<td>64-byte packet</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Power Management</strong></td>
<td>ACPI-compatible</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Signaling</strong></td>
<td>1.2-V Low-Voltage Differential Signaling (LVDS) with a 100-ohm differential impedance</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Multiprocessing Support</strong></td>
<td>Yes</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Environment</strong></td>
<td>Inside the box</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Memory model</strong></td>
<td>Coherent and noncoherent</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
HyperTransport Topologies

HyperTransport technology is designed to support up to 32 devices per channel and can mix and match components with different link widths and speeds.

Daisy chain Environment

This capability makes it possible to create HyperTransport technology devices that are building blocks capable of spanning a range of platforms and Market segment.

e.g

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, 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 2. Example HyperTransport™ Technology Device Configurations
Switch Environment

- 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.

e.g.

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.
Star topology

Whereas daisy chain configurations offer linear bus topologies much like a network “backbone,” and switch topologies expand these into parallel chains, a star topology approach that distributes HyperTransport technology links in a spoke fashion around a central host or switch offers a great deal of flexibility.
Multiprocessor Implementation
Flexible I/O Architecture

Conceptually, the architecture of the HyperTransport I/O link can be mapped into five different layers, whose structure is similar to the Open System interconnection (OSI) reference model.

<table>
<thead>
<tr>
<th>Layer</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Session Layer</strong></td>
</tr>
<tr>
<td><strong>Transport Layer</strong></td>
</tr>
<tr>
<td><strong>Protocol Layer</strong></td>
</tr>
<tr>
<td><strong>Data Link Layer</strong></td>
</tr>
<tr>
<td><strong>Physical Layer</strong></td>
</tr>
</tbody>
</table>
Physical Layer

Physical Layer is concerned with the physical connection among the various component with in the computer.

Each HyperTransport link consists of two point-to-point unidirectional data path.

- Data path widths of 2, 4, 8, and 16 bits can be implemented either upstream or downstream, depending on the device-specific bandwidth requirements.
- Commands, addresses, and data (CAD) all use the same set of wires for signaling, dramatically reducing pin requirements.

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.
Hypertransport

A point-to-point parallel bus like HyperTransport has many advantages over shared bus structures. It needs far fewer sideband signals because its enhanced 1.2V LVDS signals are not multiplexed, use less power, exhibit better noise immunity and require no passive filtering. The electrical characteristics of the link are simplified and enable much faster clock speeds and correspondingly greater bandwidth.
Figure 4. Enhanced Low-Voltage Differential Signaling (LVDS)
Data link Layer

The data link layer concerned with

- initialization and configuration sequence, periodic
- cyclic redundancy check (CRC),
- disconnect/reconnect sequence,
- information packets for flow control and error management, and
- double word framing for other packets
Protocol Layer

• The protocol layer includes the commands, the virtual channels in which they run, and the ordering rules that govern their flow.

Transport Layer

• The transaction layer uses the elements provided by the protocol layer to perform actions, such as read request and responses.
Session Layer

The session layer includes link width optimization and link frequency optimization along with interrupt and power state capabilities.

• link width optimization

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.

• link frequency optimization

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.
Hypertransport Packet format

HyperTransport Packet Format

- Header: 8- or 12 bytes
- DATA: 4-64 bytes

TRANSACTION/DATA LINK LAYER
8-12 bytes overhead

PHYSICAL LAYER
0% overhead
<table>
<thead>
<tr>
<th>Signal Name</th>
<th>Description</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>CAD</td>
<td>Commands, Addresses and Data: Carries command, address, or data information.</td>
<td>CAD width can be different in each direction.</td>
</tr>
<tr>
<td>CTL</td>
<td>Control: Used to distinguish control packets from data packets.</td>
<td></td>
</tr>
<tr>
<td>CLK</td>
<td>Clock: Forwarded clock signal.</td>
<td>Each byte of CAD has a separate clock signal. Data is transferred on each clock edge.</td>
</tr>
<tr>
<td>PWROK</td>
<td>Power OK: Power and clocks are stable.</td>
<td>Single-ended.</td>
</tr>
<tr>
<td>RESET#</td>
<td>HyperTransport Technology Reset: Resets the chain.</td>
<td>Single-ended.</td>
</tr>
<tr>
<td>LDTSTOP#</td>
<td>HyperTransport Technology Stop: Enables and disables links during system state transitions.</td>
<td>Used in systems requiring power management. Single-ended.</td>
</tr>
<tr>
<td>LDTREQ#</td>
<td>HyperTransport Technology Request: Requests re-enabling links for normal operation.</td>
<td>Used in systems requiring power management. Single-ended.</td>
</tr>
</tbody>
</table>
HyperTransport Control Packets

Base control packets are generally 4 to 8 byte long (in case of 64-bit addressing they are 12 byte long).

It is divided into three types:

- Request Packet → Sized read, write, flush, fence, broadcast message, atomic read modify
- Response Packet → Read, Target Done
- Info Packet → NOP, Sync/Error
Applications for HyperTransport

• Front-Side Bus Replacement

The primary use for HyperTransport is to replace the front-side bus, which is currently different for every type of machine. Computer implemented with HyperTransport is faster and more flexible. A single PCI↔HyperTransport adaptor chip will work with any HyperTransport enabled microprocessor and allow the use of PCI cards with these processors.

• Multiprocessor interconnect

Another use for HyperTransport is as an interconnect for NUMA multiprocessor computers. AMD uses HyperTransport with a proprietary cache coherency extension as part of their Direct Connect Architecture in their Opteron and Atholon 64FX (DCA) The HORUS interconnect from Newisys extends this concept to larger clusters.
• **Router or Switch Bus Replacement**

  HyperTransport can also be used as a bus in routers and switches. Routers and switches have multiple connections ports and data has to be forwarded between these ports as fast as possible.

• **HTX and Co-processor interconnect**

  The issue of bandwidth between CPUs and co-processors has usually been the major stumbling block to their practical implementation. After years without an officially recognized one, a connector designed for such expansion using a HyperTransport interface was introduced and is known as HyperTransport expansion (HTX).
Conclusion

• HyperTransport technology is a powerful board-level architecture that delivers high bandwidth, low latency, scalability, PCI compatibility, and extensibility.

• When processor-native, HyperTransport provides an integrated front-side bus that eliminates custom/proprietary buses and the need for additional glue logic.

• As an integrated I/O bus, HyperTransport eliminates the need for multiple local buses, ultimately simplifying overall system design and implementation.

• As a link to legacy PCI and PCI-X subsystems, HyperTransport extends the life of those popular systems and as a emerging, high-speed serial buses such as PCI Express, it becomes a board-level system architecture