HyperTransport Protocol

By Ashkan Tariverdi

TABLE OF CONTENTS

HyperTransport Overview

The HyperTransport protocol is:

- **High-speed:** up to 51.2 GB/s in some versions.
- **Low-latency**
- **Point-to-point interconnect**

HyperTransport Usage

- **Processor Interconnect:** It connects multiple processors in multi-core or multi-processor systems, enabling high-speed communication between them.

- **Memory and Peripheral Communication:** It is also used to interconnect memory controllers, network adapters, and other peripheral devices.

- **Server and High-Performance Systems:** It’s commonly found in servers, workstations, and high-performance computing systems where low latency and high bandwidth are critical.

HyperTransport Versioning

HT version	Year	Max. freq.	Max. link width	Max. aggregate bandwidth (GB/s)		
				bi-directional	16-bit unidirectional	32-bit unidirectional
1.0	2001	800 MHz	32-bit	12.8	3.2	6.4
1.1	2002	800 MHz	32-bit	12.8	3.2	6.4
2.0	2004	1.4 GHz	32-bit	22.4	5.6	11.2
3.0	2006	2.6 GHz	32-bit	41.6	10.4	20.8
3.1	2008	3.2 GHz	32-bit	51.2	12.8	25.6

HyperTransport Physical Layer

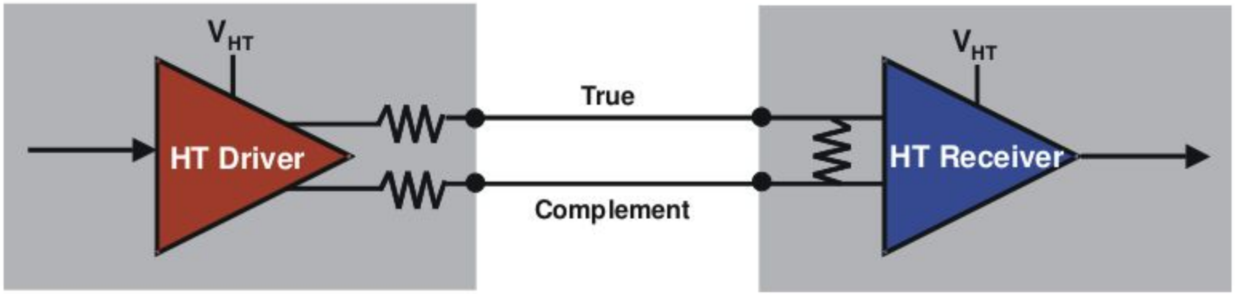
- **Differential Signaling:** HyperTransport uses differential signaling to transmit data, meaning two wires carry the signals in opposite polarity.

- **Encoding**: The protocol employs 8b/10b encoding to ensure reliable data transmission.

- **Signal Integrity**: As HT is designed for high-speed data transfer, special attention is given to minimizing signal degradation. Features like termination (to prevent signal reflections) and voltage swing control are used to ensure the stability of the data signal.

Differential Signaling

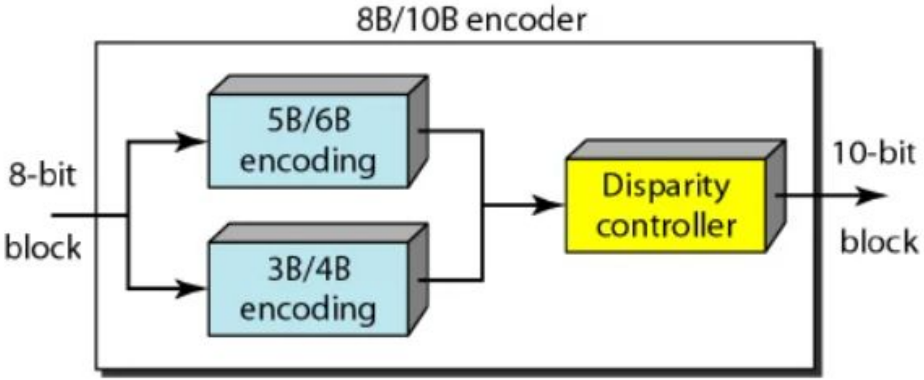
It helps minimize noise and crosstalk between wires, which is critical at high speeds, reducing the risk of data corruption. It is especially effective in environments with a lot of electrical noise, such as servers or other high-performance computing environments.



Encoding

8b/10b Encoding maps every 8-bit data unit to a 10-bit symbol. This encoding scheme serves two important purposes:

- **DC Balance**: It ensures that there is no significant DC bias in the signal, which is essential for differential signaling.
- **Error Detection**: It provides a form of built-in error detection, as some codewords are not valid, allowing the receiver to detect transmission errors.



HT Serial Communication

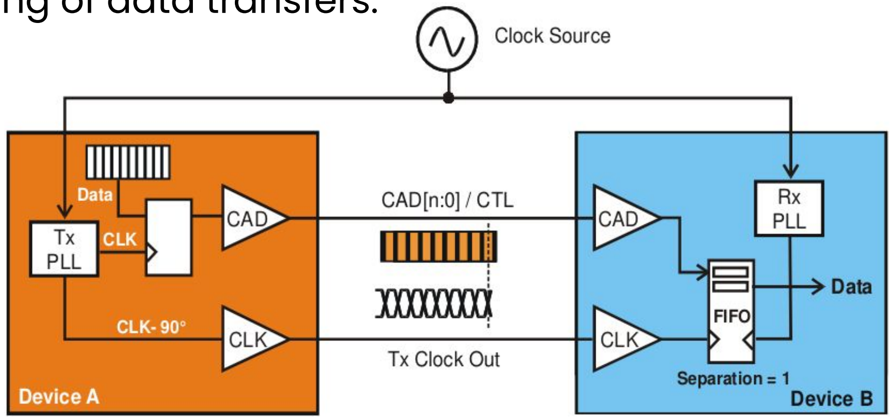
Serial Communication means that data is transmitted bit by bit over a single channel.

- Signal Integrity
- Scalability over long distances
- High-speed
- Reduces complexity
- Reduces physical limitations of parallel systems, such as skew and crosstalk

Synchronous Transmission

Device A (Transmitter):

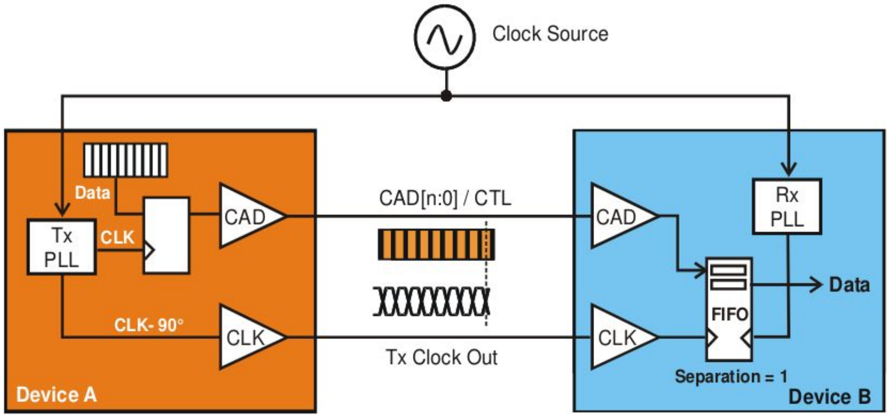
- **Tx PLL (Phase-Locked Loop)**: Device A generates a **Tx clock** which is synchronized with the main clock.
- The **CAD** (Command Address Data) signal group carries the address, control signals, and data over the bus.
- The **CLK** signal indicates the timing of data transfers.



Synchronous Transmission (Cont'd)

Device B (Receiver):

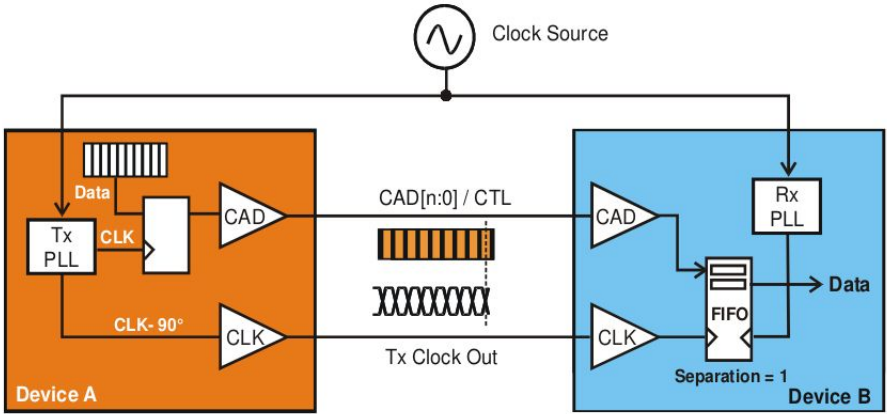
- The **Rx PLL** in Device B locks onto the clock signal sent by Device A, ensuring proper reception of the data.
- Device B receives the data using the synchronized clock signal.
- The **FIFO** buffer is used to store the received data temporarily before it is processed.



Synchronous Transmission (Cont'd)

Clock Phase

- The clock is split into two phases: **CLK** and **CLK-90°**. The 90-degree phase shift ensures that the transmitter and receiver operate in **half-clock cycle synchronization.**



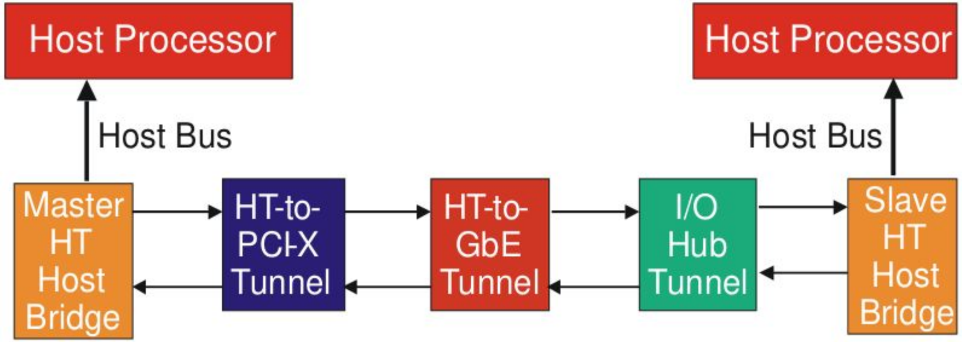
Double-Hosted Chain Conf.

Master HT Host Bridge

- It is responsible for managing the data flow from the first host processor to other connected devices via the bus.

Slave HT Host Bridge

- It is responsible for managing the data flow from the second host processor and serves as the **slave** in this portion of the chain.



Double-Hosted Chain Conf. (Cont'd)

HT-to-PCI-x Tunnel

- This **tunnel** acts as an intermediary device that enables communication between the HyperTransport network and a **PCI-X** bus

HT-to-GbE Tunnel

- This **tunnel** facilitates communication between the HyperTransport bus and **Gigabit Ethernet (GbE)** devices.

I/O Hub Tunnel

- The **I/O Hub Tunnel** connects to other peripheral devices and serves as an intermediary between the **HyperTransport bus** and I/O devices.

Packet-Based Protocol

In **HyperTransport**, the **packet-based protocol** involves several different packet types, each with a specific role in managing and transferring data.

- **Information Packets:** These packets are used for communication between devices and they are **4 bytes in size**.

  Example:

  - **NOP (No Operation)**: A packet that indicates the idle condition or is used for synchronization without any further data.
  - **Sync/Error Packet**: This packet is used for synchronization or error reporting.

Packet-Based Protocol (Cont'd)

- **Request Packets:** These packets initiate transactions such as **read** or **write** operations. **Request packets** can vary in size, typically 4 or 8 bytes depending on whether an address field is required.

  Example:

  - **Sized Read**: This packet initiates a read operation from a target device. It specifies the address from which data should be read.
  - **Posted Writes**: Data immediately follows the write request and is not acknowledged with a response packet.
  - **Broadcast Message**: This is a special packet type used to broadcast a message to all devices on a link.

Command Type Code
Unit ID
Start Address
Mask/Count
Compatibility Flag

Packet-Based Protocol (Cont'd)

- **Response Packets:** These packets are used to respond to requests and can contain data or simply acknowledge the completion of a transaction.

  Example:

  - **Target Done**: Sent in response to a non-posted write request, indicating that the operation has been completed.
  - **Data Response**: This is used to send back the data requested by a **Sized Read** request.

Command Type Code
Data Payload
Status

Packet-Based Protocol (Cont'd)

- **Data Packets:** These packets are used to carry the actual data payload, typically in response to a request. They can carry up to **64 bytes** of data.

  Example:

  - **Data Payload**: This is the actual data transferred between devices during a read or write operation.

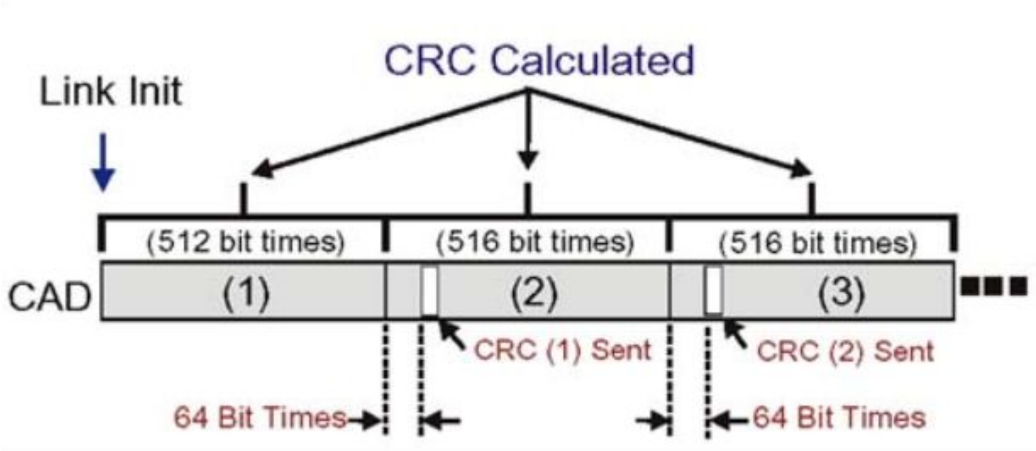
Data
Mask/Count

Flow Control

- **Packets Never Start Unless Completion Assured:** No link transmitter ever starts a packet transfer unless it is known the packet can be accepted by the receiver. This is accomplished with the "coupon based" flow control scheme.
- **Transfer Length Is Always Known:** Hypertransport control packets have a fixed size (four or eight bytes) and data packets have a *known* and *maximum* transfer length. This makes buffer sizing and flow control much more straightforward as both transmitter and receiver are aware of their actual transfer commitments.

Error Detection

Cyclic Redundancy Check (CRC): HyperTransport uses **CRC checksums** for error detection in its packets. Each packet transmitted across the system includes a **CRC field** that is calculated based on the content of the packet.



RESOURCES

Main

- HyperTransport System Architecture, by Don Anderson & Jay Trodden

Additional

- https://en.wikipedia.org/wiki/HyperTransport
- https://www.rfwireless-world.com/Terminology/Advantages-and-Disadvantages-of-8B-10B-encoding.html
- https://flylib.com/books/en/3.356.1.81/1/

Thank you for your attention!