



## پروتکل HyperTransport

استاد: دکتر فصحتی

اشکان تارپوردی ۴۰۱۱۰۵۷۵۳

## معرفی پروتکل HyperTransport

پروتکل HyperTransport یک پروتکل پرسرعت، تاخیر کم، اتصال نقطه به نقطه است که برای متصل کردن پردازنده‌ها، حافظه و سایر دستگاه‌های جانبی درون یک سیستم کامپیوتری، استفاده می‌شود. این پروتکل به منظور تسریع و جایگزینی برای اتصالات گذرگاه مشترک (Shared Bus) طراحی شده است.

## کاربرد پروتکل HyperTransport

- **اتصال پردازنده‌ها:** در سیستم‌های چند پردازنده‌ای (multi-processor) و چند هسته‌ای (multi-core)، برای اتصال پردازنده‌ها استفاده می‌شود. این پروتکل، سرعت بالا در انتقال پیام‌ها بین پردازنده‌ها را فراهم می‌کند.
- **اتصال حافظه و دستگاه‌های جانبی:** از این پروتکل برای اتصال کنترلرهای حافظه، آداپتورهای شبکه و سایر دستگاه‌های جانبی، استفاده می‌شود.
- **سرور و سیستم‌های HPC:** این پروتکل در سرورها و محیط‌های High-performance computing که تاخیر و سرعت و پهنای باند اهمیت بسیار بالایی دارد، استفاده می‌شود.

## چرایی طراحی HyperTransport

نیاز به یک پروتکل پرسرعت و بهینه‌تر برای مقابله با محدودیت‌های پهنای باند Front-Side Bus های سنتی، باعث طراحی این پروتکل شد. با ظهور پردازنده‌های چند هسته‌ای و افزایش تقاضا برای انتقال داده پرسرعت، نیاز به یک پروتکل توسعه‌پذیر و با پهنای باند بالا، بیش از قبل احساس شد.

## ورژن‌های HyperTransport

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

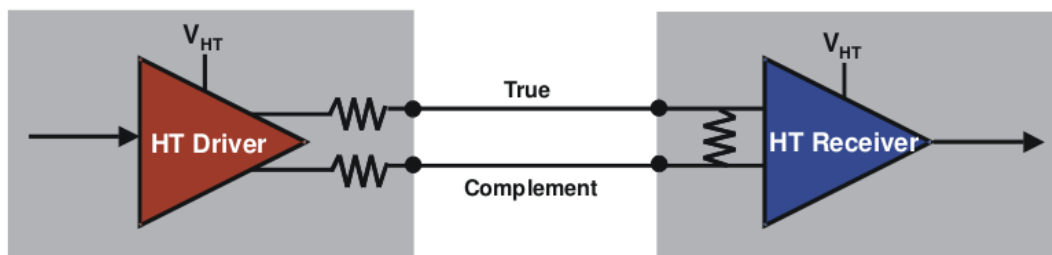
3.0	2006	2.6 GHz	32-bit	41.6	10.4	20.8
3.1	2008	3.2 GHz	32-bit	51.2	12.8	25.6

نکات اصلی در رابطه با ویژگی‌های هر ورژن:

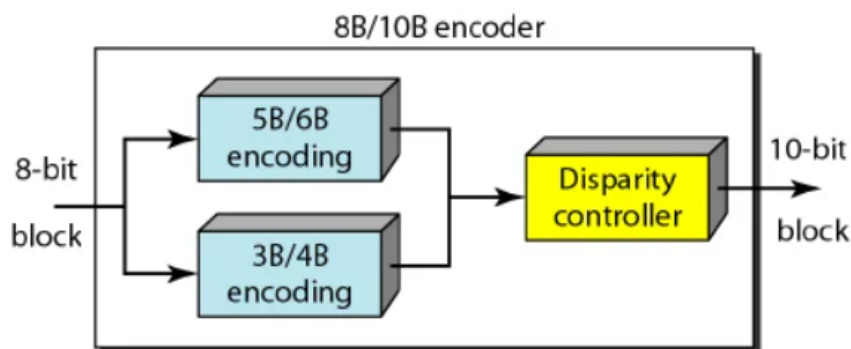
- **پهنای باند:** هر ورژن نسبت به ورژن قبلی، پهنای باند بالاتری را ارائه می‌کند.
- **توسعه‌پذیری:** پروتکل HyperTransport بین ۱ تا ۱۶ Lane را ساپورت می‌کند.
- **تشخیص خطا:** مکانیزم‌های تشخیص خطا پیشرفته در ورژن‌های جدیدتر ارائه شده است تا reliability را افزایش دهد.

## لایه‌ی فیزیکی HyperTransport

- لایه‌ی فیزیکی پروتکل HyperTransport، پایین‌ترین لایه در این پروتکل است. این لایه، با انتقال داده‌های واقعی بر روی یک رسانای فیزیکی، برای مثال سیم مسی یا فیبر نوری، سر و کار دارد. ویژگی‌های اصلی این لایه عبارتند از:
- **سیگنالینگ تفاضلی (Differential Signaling):** این پروتکل از سیگنالینگ تفاضلی برای انتقال داده استفاده می‌کند. سیگنالینگ تفاضلی یعنی دو سیم سیگنال‌ها را با قطبیت مخالف ارسال می‌کنند. این تکنیک به کاهش نویز و تداخل‌های الکترومغناطیسی و افزایش یکپارچگی سیگنال به خصوص در مسافت‌های طولانی، کمک می‌کند.



- **کدگذاری:** این پروتکل از انکودینگ 8B/10B استفاده می‌کند. این انکودینگ، قابلیت اطمینان برای انتقال داده را افزایش می‌دهد. این انکودینگ، داده‌ی ۸ بیتی را به نمادهای ۱۰ بیتی map می‌کند. این تکنیک باعث می‌شود که DC Balance بودن پروتکل رعایت شود و نیاز به DC Component نداشته باشیم. همچنین باعث کاهش Jitter می‌شود که برای حفظ یکپارچگی داده برای انتقال داده‌ها بر روی رسانه‌های پرسرعت لازم است.



- یکپارچگی سیگنال: از آنجایی که این پروتکل برای سرعت بالا طراحی شده است، توجه خاصی به مینیمم کردن تنزل سیگنال (Signal Degradation) شده است. ویژگی‌های چون Termination برای جلوگیری از انعکاس سیگنال و Voltage swing control برای اطمینان از پایداری از سیگنال‌ها، پیاده‌سازی شده است.

### اتصالات اختیاری پروتکل HyperTransport

- **سوییچ‌ها (Switch):** در سیستم‌های پیچیده (مانند سرورهای بزرگ یا پیکربندی‌های چند گره‌ای)، از سوییچ‌ها برای اتصال دستگاه‌های متعدد از پروتکل HyperTransport استفاده می‌شود. این سوییچ‌ها ضروری نیستند ولی برای اتصالات در سیستم‌ها توسعه‌پذیر مفید هستند.
- **دستگاه‌های جانبی:** در برخی از سیستم‌ها، دستگاه‌های خارجی اضافی می‌توانند به کمک این پروتکل به سیستم متصل شوند.

### اتصال سریال (Serial)

پروتکل HyperTransport از ارتباط سریال به جای موازی (Parallel) استفاده می‌شود. اتصال سریال یعنی داده‌ها بیت به بیت بر روی کانال ارتباطی ارسال می‌شوند. این تکنیک در خلاف ارسال داده‌ها به صورت موازی یا parallel است. در اتصال موازی، چندین بیت به صورت همزمان بر روی چندین کانال ارسال می‌شوند. این پروتکل از ارتباطات سریال استفاده می‌کند تا یکپارچگی داده و توسعه‌پذیری بر روی مسافت‌های زیاد و سرعت بالا را حفظ کند. ارتباط سریال، پیچیدگی را کاهش می‌دهد و محدودیت‌های فیزیکی سیستم‌های موازی مانند skew و crosstalk را ندارد.

### تولید و انتقال سیگنال

- **تولید سیگنال:** پروتکل HyperTransport، سیگنال‌ها را به وسیله‌ی تبدیل داده (انکودینگ 8B/10B) به کمک سیگنالینگ تفاضلی (دو سیم که سیگنال‌های متضاد را حمل می‌کنند) تولید می‌کنند. پس از آن، سیگنال‌ها بر روی یک رسانه فیزیکی مانند PCB traces یا فیبر نوری به مقصد منتقل می‌شوند.

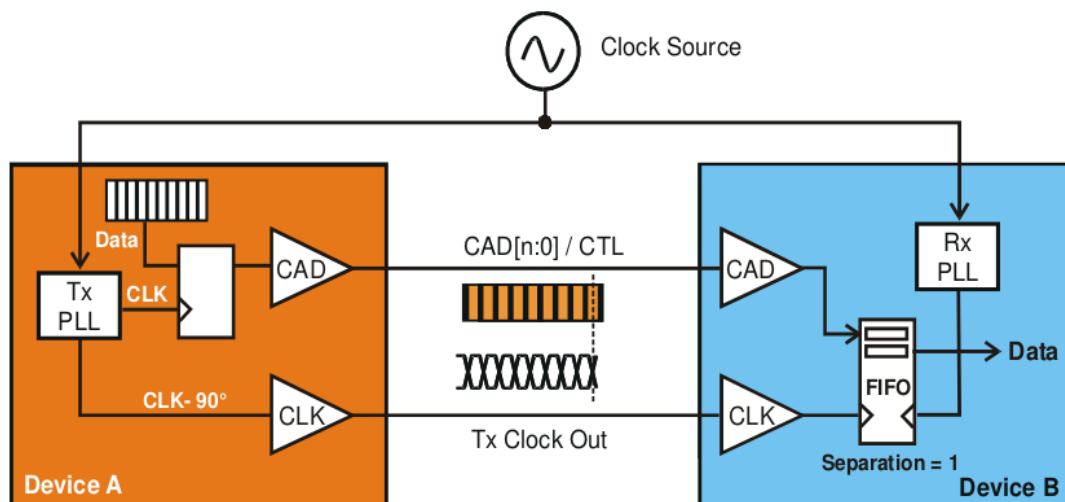
- **انتقال داده:** از آنجایی که پروتکل HyperTransport از سیگنالینگ تفاضلی استفاده می‌کند، دو سیم با قطبیت متضاد برای ارسال داده استفاده می‌شود. گیرنده، تفاضل ولتاژ سیگنال‌های دریافتی را مقایسه می‌کند و داده را استخراج می‌کند.

## انتقال همزمان (Asynchronous)

انتقال داده در این پروتکل به صورت همزمان است. گیرنده‌ها و فرستنده‌ها، با یک کلاک یکسان همزمان شده‌اند. گیرنده‌ها و فرستنده‌ها با توجه به کلاک یکسان، از ارسال و دریافت داده‌ها در بازه‌های زمانی یکسان و خاص مطمئن می‌شود. این تکنیک برای ارسال داده‌ها با سرعت بالا برای یکپارچگی داده، ضروری است. ویژگی همزمانی پروتکل HyperTransport اجازه می‌دهد که داده‌ها با نرخ بالا ارسال شوند و خطایی رخ ندهد. انتقال داده‌ها در پروتکل HyperTransport به صورت همزمان و به صورت زیر است:

دستگاه A (فرستنده):

- Tx PLL: دستگاه A یک Tx کلاک تولید می‌کند که با کلاک اصلی هم‌گام شده است.
  - CAD: گروه سیگنالی Command Address Data سیگنال‌های کنترلی، آدرس و داده را بر روی گذرگاه، حمل می‌کنند.
  - CLK: سیگنال کلاک زمان ارسال داده را مشخص می‌کند.
- دستگاه B (گیرنده):
- Rx PLL: دستگاه B بر اساس سیگنال کلاکی که توسط دستگاه B تولید شده است، سیستم را قفل کرده تا از دریافت درست داده‌ها اطمینان حاصل کند.
  - FIFO buffer: در سمت گیرنده یک بافر First in First out قرار دارد که داده‌ها قبل از پردازش در آن ذخیره می‌شوند.



## پی‌کربندی HyperTransport

پی‌کربندی پروتکل HyperTransport از روشی به نام Double-Hosted Chain استفاده می‌کند. این پی‌کربندی بخش‌های متعددی دارد که عبارتند از:

- **Master HT Host Bridge**: این پل (Bridge) برای برقراری ارتباط و مدیریت جریان داده از پردازنده میزبان اول به سایر دستگاه‌ها از طریق گذرگاه، است.
- **Slave HT Host Bridge**: این پل وظیفه برقراری ارتباط و مدیریت جریان داده از پردازنده میزبان دوم به سایر پردازنده‌ها و دستگاه‌ها به عنوان slave را دارد.
- **HT-to-PCI-X Tunnel**: این تونل به عنوان یک دستگاه میانی عمل می‌کند و برای برقراری ارتباط گذرگاه PCI و HyperTransport است.
- **HT-to-GbE Tunnel**: این تونل به عنوان یک دستگاه میانی عمل می‌کند و برای برقراری ارتباط گذرگاه اینترنت‌های سرعت بالا مانند Gigabit Ethernet و HyperTransport است.
- **I/O Hub Tunnel**: این تونل به عنوان یک دستگاه میانی بین سایر دستگاه‌های جانبی و HT عمل می‌کند.

## انواع بسته‌ها در پروتکل HyperTransport

- این پروتکل به جای ارسال پیام‌ها به صورت عادی، از ارسال پیام‌ها به روش بسته‌ای استفاده می‌کند. این پروتکل انواع مختلفی از بسته‌ها برای عملیات مختلف را شامل می‌شود. به بررسی این بسته‌ها می‌پردازیم:
- **بسته‌های اطلاعاتی (Information Packet)**: این بسته به منظور برقراری ارتباط بین دستگاه‌ها است و معمولاً ۴ بایت سائز آن است. برای مثال:
    - **NOP**: این بسته به منظور شرایط بیکار یا Idle است و برای همگام‌سازی استفاده می‌شود.
    - **Sync/Error**: از این بسته به عنوان همگام‌سازی یا پیام خطاها استفاده می‌شود.
  - **بسته‌های درخواست (Request Packet)**: این بسته‌ها، transactionها مانند خواندن و نوشتن را برقرار می‌کنند. بسته‌های درخواست می‌توانند در سائز (معمولاً ۴ یا ۸ بایت) متفاوت باشند. برای مثال:
    - **Sized Read**: این بسته‌ها یک عملیات خواندن از دستگاه target را انجام می‌دهند. این بسته آدرسی که خواندن از آن انجام می‌شود را مشخص می‌کند.
    - **Posted Writes**: داده بعد از درخواست نوشتن، ارسال می‌شود و با یک پاسخ ACK همراه می‌شود.
    - **Broadcast Message**: این بسته‌ها برای broadcast استفاده می‌شود و یک پیام را به تمامی دستگاه‌ها ارسال می‌کنند.
  - **بسته‌های پاسخ**: این بسته‌ها برای پاسخ دادن به درخواست‌ها استفاده می‌شوند و می‌توانند شامل داده باشند. برای مثال:

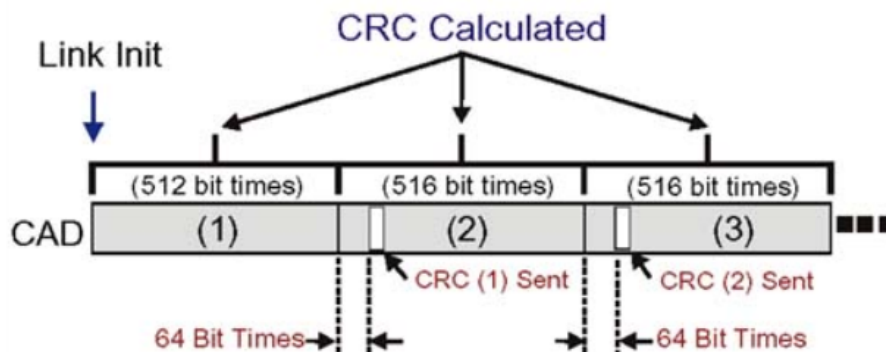
- Target Done: در جواب یک درخواست نوشتن non-posted ارسال می‌شود و نشان می‌دهد که عملیات به درستی انجام شده است.
- Data Response: این پیام به عنوان ارسال داده در جواب یک درخواست Sized Read استفاده می‌شود.
- بسته‌های داده (Data Packet): این بسته‌ها به عنوان حامل‌های داده در جواب درخواست‌ها ارسال می‌شوند. این بسته‌ها تا ۶۴ بایت داده را می‌توانند حمل کنند. برای مثال:
- Data Payload: این بسته در واقع داده واقعی است که در عملیات‌های خواندن یا نوشتن منتقل می‌شود.

### مدیریت جریان داده (Flow Control)

- مدیریت جریان داده در این پروتکل در دو حالت انجام می‌شود که عبارتند از:
- بسته‌ها تا زمانی که از ارسال کامل بسته‌های قبلی اطمینان حاصل نکنند، داده را ارسال نمی‌کنند. هیچ ارسال‌کننده‌ای، شروع به ارسال بسته نمی‌کند تا زمانی که مطمئن نشود بسته‌ی قبلی توسط گیرنده دریافت شده است. این تکنیک در قالب coupon based معرفی می‌شود.
  - سائز انتقال همواره مشخص است. بسته‌های کنترلی پروتکل HT همواره سائز ثابتی دارند و توسط گیرنده و فرستنده مشخص هستند. این مشخصه، به مدیریت جریان داده کمک می‌کند و پیاده‌سازی آن را راحت‌تر می‌کند.

### تشخیص خطا (Error Detection)

تشخیص خطا در این پروتکل بر مبنای Cyclic Redundancy Check است. این پروتکل از CRC به منظور تشخیص خطا استفاده می‌کند. هر بسته که ارسال می‌شود، یک فیلد مخصوص CRC دارد که بر اساس محتوای شبکه محاسبه می‌شود و تشخیص خطا بر اساس آن انجام می‌شود.



### تصحیح خطا (Error Correction)

تکنیکی مبنا بر تصحیح خطا در این پروتکل وجود ندارد و تنها از Retransmission برای بسته‌هایی که دچار خطا شده‌اند، استفاده می‌شود. بنابراین مکانیزم جدایی برای تصحیح خطا در این پروتکل، وجود ندارد.