

# Next-Generation Data Transport over Metro Area Networks

The Xilinx GFP core enables efficient transport of LAN/SAN protocols over SONET-based networks.

by Tom Fischaber  
Sr. Design Engineer, IP Solutions Division  
Xilinx, Inc.  
[tom.fischaber@xilinx.com](mailto:tom.fischaber@xilinx.com)

Krista Marks  
Sr. Manager, IP Solutions Division  
Xilinx, Inc.  
[krista.marks@xilinx.com](mailto:krista.marks@xilinx.com)

Hamish Fallside  
Sr. Manager, Advanced Products Division  
Xilinx, Inc.  
[hamish.fallside@xilinx.com](mailto:hamish.fallside@xilinx.com)

SONET networks are ubiquitous in the telecommunications industry for transporting both voice and data over long distances. Standard protocols have targeted the underlying transport layer, including ATM for voice and data and HDLC or PPP for data transfer. However, each of these protocols introduces bandwidth inefficiencies, as none were developed specifically to address data transport over SONET/SDH networks.

Additionally, telecom carriers are motivated to increase revenues by diversifying the types of client traffic transported across their networks and optimizing their bandwidth utilization. This includes capturing new market sectors such as storage area networks (SAN – utilizing Fibre Channel) and emerging video on demand (utilizing DVB-ASI). In this case, Fibre Channel and DVB represent two specific types of client traffic or client data as defined by these data network protocols.

The Generic Framing Procedure (GFP) is the first encapsulation mechanism capable of addressing this wide range of data transport applications by supporting a suite of client network protocols (summarized in Table 1). GFP was introduced by the International Telecommunication Union (ITU) as recommendation G.7041/Y.1303, and provides a flexible, efficient mapping of various protocols onto a transport network.

In this article, we'll describe a flexible suite of networking solutions that address the needs of systems vendors deploying metro equipment to provide these services. Xilinx supports applications ranging from a

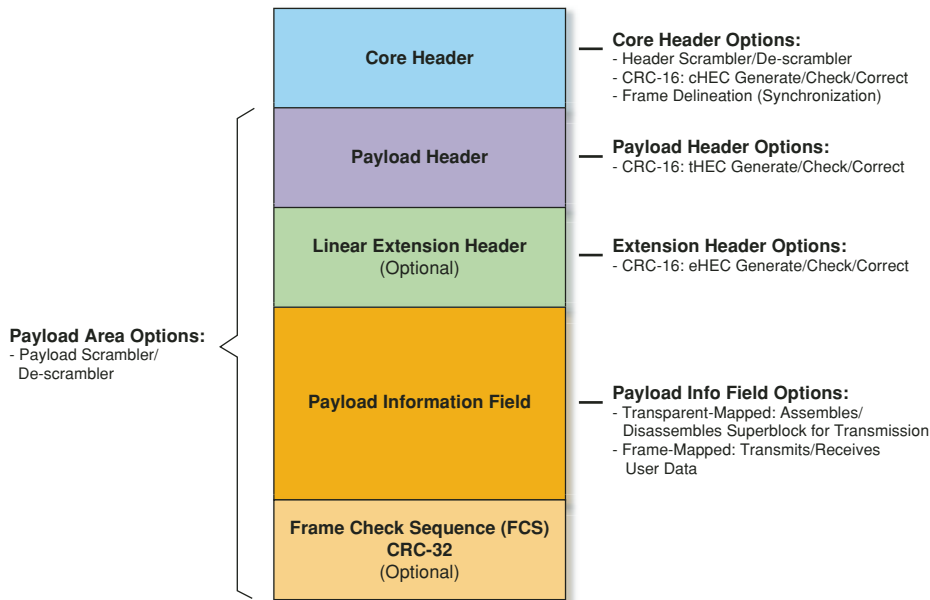


Figure 1 – GFP core options for frame processing

completely integrated client adaptation with Virtex-II Pro™ devices to low-cost protocol encapsulation with Spartan-3™ FPGAs. The multi-client protocol support in Virtex-II Pro and Virtex-II Pro X high-speed RocketIO™ transceivers enables seamless communication with protocols operating at up to 10.3125 Gbps. Coupling this technology with the

provides a fully configurable way to implement custom solutions that can dynamically adapt in a rapidly changing environment.

### The GFP Specification

The GFP uses an octet-based stream of data that maps directly into an octet-synchronous stream, such as synchronous optical network/synchronous digital hierarchy (SONET/SDH). GFP frames are scrambled to ensure DC balance and running disparity, and they are delineated by using a length field within the core header, as shown in Figure 1.

Because the start and end of the frame is embedded within the GFP stream, synchronization of the two GFP end-points must first occur to ensure that data can be transmitted. Synchronization at an individual end-point is

achieved by the detection of a correct CRC over the core header, then using the length field to point to the start of the following frame. Once this process is successfully repeated a programmable number of times, the GFP stream is synchronized, and data can be transmitted.

Some aspects of the GFP protocol are common to all implementations. This includes options such as frame delineation and synchronization; CRC insertion/detection/correction; and scrambling.

In addition to the common aspects of the GFP protocol, client-specific functions are required to handle unique differences in protocol mapping. This includes options specific to the two types of client data mapping: frame-mapped (GFP-F) and transparent-mapped (GFP-T). Table 1 lists all of the GFP-F and GFP-T protocols supported in the G.7041 specification.

GFP-F supports variable-sized packet lengths of framed data, where one client frame (such as an Ethernet frame) maps directly into one GFP-F frame. This requires a media access controller (MAC) in the system to terminate the Layer 2 protocol. In Ethernet, for example, an Ethernet MAC removes the preamble and start of frame delimiter, checks the CRC, and passes the Ethernet frame to the GFP end-point for encapsulation.

GFP-T supports fixed-sized packet lengths and transports block-coded constant rate bitstreams (such as Fibre Channel, Ethernet, or ESCON/SBICON). This generates a GFP frame that encapsulates block-coded data, which contains the client protocol 8B10B data and control (symbols) that are mapped to 64B65B block codes.

The transparent-mapped protocol does not require that application buffers complete frames before transmission. Instead, both data and control symbols are accumulated. Eight 8B/10B symbols (plus a flag bit) are combined to create a 64B/65B block code. This block code will include both data words and control characters.

Eight 64B/65B block codes are then combined to create superblocks (65 bytes of data + CRC16). Multiple superblocks are combined to create the GFP payload, where the number of superblocks per frame is protocol-specific (95 for Gigabit Ethernet and 13 for Fibre Channel). GFP-T does not require MAC functionality, as it is notionally transparent to the protocol transmitted.

The selection of GFP-F versus GFP-T depends on the application and system requirements. GFP-F provides bandwidth

FRAME MAP PROTOCOLS	TRANSPARENT MAP PROTOCOLS
Ethernet	Fibre Channel
PPP	Gigabit Ethernet
RPR (IEEE 802.17)	ESCON
FC-BBW	DVB ASI
Multiple-Access Protocol Over SDH (MAPOS)	FICON
	Asynchronous FC

Table 1 – Client network protocols supported by GFP

embedded PowerPC™ processor enables not only client adaptation, but also real-time control and processing capabilities within a single FPGA.

Implemented on these platforms and combined with a suite of additional Xilinx intellectual property, the new GFP core

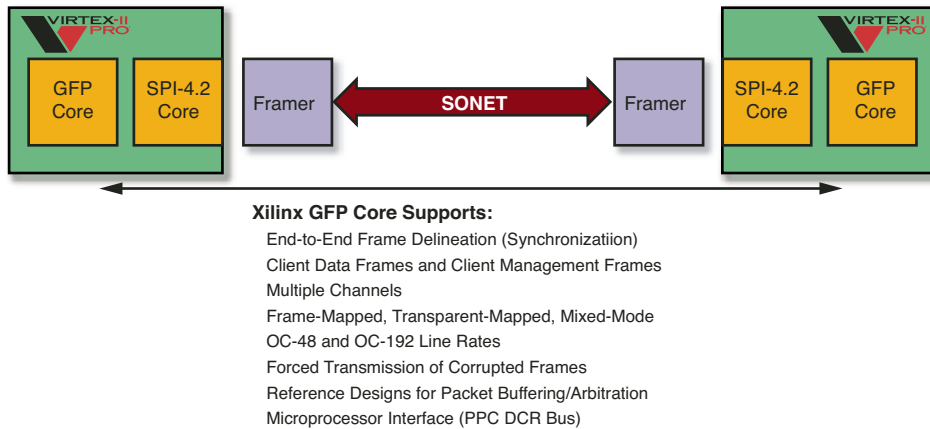


Figure 2 – Features supported by the Xilinx GFP IP core, enabling system solutions

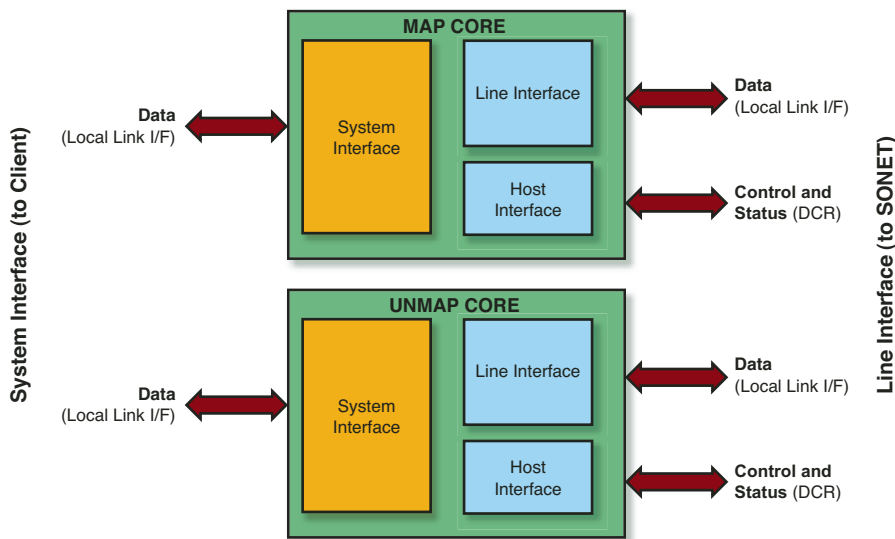


Figure 3 – Block diagram of GFP IP MAP and UNMAP core interfaces

efficiency by ensuring that only actual data is transmitted, whereas GFP-T transmits all information including data, framing codes, preamble, and idles.

GFP-F incurs higher latency through the system, because complete frames must be buffered before transmission. This may imply the use of external memory, depending on the system.

GFP-T does not require complete frame transmission, and therefore can achieve lower system latencies. In practice, transmission over long distances will incur a latency (due to the medium) that requires additional functionality in the client adaptation layer to ensure that the protocols' latency requirements are met. This is called

“spoofing” and is critical for some client protocols, such as Fibre Channel, where strict latency requirements exist.

### Xilinx GFP Core

Xilinx offers a GFP IP core solution that supports all of these protocols and fully implements the G.7041/Y.1303 specification defined by the ITU-T. This includes system-level capability of end-to-end frame delineation, support for both client management and data frames, and configuration of frame or transparent mapping on a per-channel basis. In addition to supporting all of the features specified by G.7041, the Xilinx solution also provides options to facilitate system-level integration and debugging.

With the Xilinx CORE Generator™ tool, you can easily configure the GFP core to support your system requirements, selecting one of three operating modes: frame-mapped, transparent-mapped, or mixed mode. Mixed mode enables you to specify on a per-channel basis whether the channel is frame- or transparent-mapped. You can also change the mode through the GFP host interface.

The core also provides optional support for linear extension headers, which can be configured for as many as 10 unique channels. You can select an optional generic host interface that enables you to modify the control and status registers in real time. Some of the key system features of the GFP core are highlighted in Figure 2.

Using the CORE Generator system, you can fully customize the behavior of the core. A specific configuration determines how the core processes frames and also determines the resources required for implementation. The composition of a frame and the options provided are illustrated in Figure 1.

The GFP core is split into a MAP core and an UNMAP core, as illustrated in Figure 3. The MAP core receives client network protocol data (such as Ethernet) on the system interface, encapsulates this data, and transmits the resulting frames on the line interface (such as SONET).

The UNMAP core does the same process in reverse, receiving encapsulated data on the line interface and de-mapping the frames to extract client network protocol data, which is in turn transmitted on the system interface. The GFP core utilizes the Xilinx LocalLink interface standard on the line and client interfaces for direct connection to other cores and reference designs.

In frame-mapped mode, the client supplies the MAP core's system interface with a complete data frame, which is then transmitted without pause. In the transparent mapped mode, the MAP core automatically handles underflow by inserting pad words into the frame so that the client does not have to provide uninterrupted frames of data.

Each core has a host interface (a PowerPC™ device control register [DCR]

bus) that provides access to a bank of control and status registers. If you only require the default control signals, you can configure the core without status registers to reduce the number of resources.

### Xilinx Complete Solutions

Xilinx provides a complete and versatile solution that allows you to configure the FPGA for your system needs. A myriad of applications exist for GFP, and the architecture and requirements of the system will drive any given set of requirements.

For example, if you are utilizing a framer that is GFP-aware, you can configure the GFP core to enhance the framer by supporting options that the framer does not. Common examples of this include channelization using the linear extension header, or transparent map mode.

If the application is using a framer that is not GFP-aware, you can configure the GFP core to perform all GFP functionality, including frame delineation and synchronization, CRC, and scrambling. Using CORE Generator software enables you to specifically craft the implementation that meets your requirements, and thus minimize the resource utilization of the FPGA.

In addition to the GFP core, other Xilinx solutions enable the implementation of a complete system to transfer Ethernet or Fibre Channel over SONET. Some of these additional solutions are illustrated in Figure 4 and described below:

- Configurable PCS (CPCS) Xilinx application note XAPP739 demonstrates how to utilize Virtex-II Pro MGTs by providing a configurable PCS for a suite of protocols (FC [1G, 2G], ESCON/SBCON, and GE). It dynamically controls the PCS mode on each port, using the PowerPC and scales to support as many as eight channels in both transparent and frame map modes.
- The System Packet Interface level 4 phase 2 (SPI-4.2) IP solution provides a fully compliant core to address the data path connectivity between POS (Packet Over SONET) physical layer

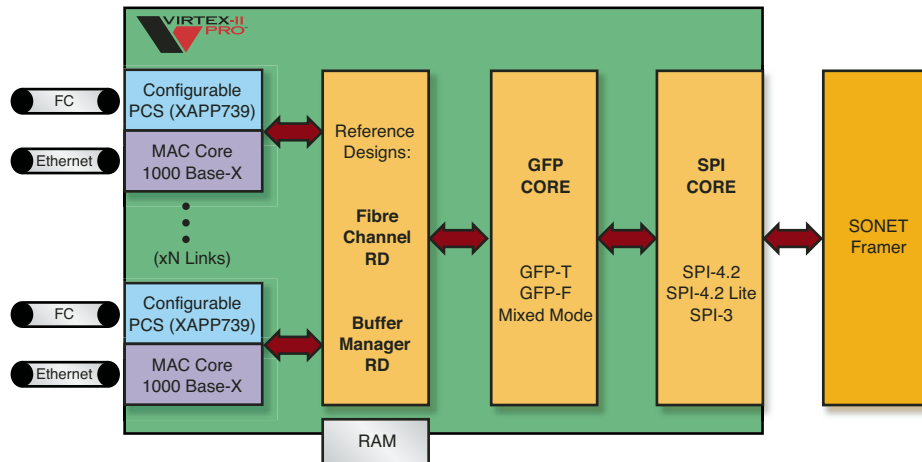


Figure 4 – A GFP application example of Ethernet over SONET and FC over SONET

devices and link layer devices in networking and communications systems. The SPI-4.2 core offers support for a resource-optimized OC-48 (2.5 Gbps) interface and a high-performance (greater than 14 Gbps) interface to connect network processors with OC-192 framers and mappers, as well as gigabit and 10 Gigabit Ethernet data-link MACs.

mechanism that address the latency-sensitive Fibre Channel requirements.

- Xilinx reference design XAPP695 demonstrates an example sub-system integration in Virtex-II Pro devices for a multi-channel Gigabit Ethernet to SPI-4.2, complete with PowerPC control plane. This topic was covered in the Summer 2004 issue of the *Xcell Journal*, in the article, “Ethernet Aggregation of GFP Framing in Virtex-II Pro.” The design does not include the core described here.

### Conclusion

The Virtex-II Pro family of FPGAs provides a powerful, flexible platform for implementing networking solutions. The high-speed serial MGTs and PowerPC, combined with proven IP cores and reference designs, provide pre-verified solutions for managing and transporting a wide variety of data protocols.

The ITU-T GFP specification provides a flexible and efficient mapping of these data protocols onto a transport network. Using Xilinx Virtex-II Pro FPGAs combined with the GFP core, you can quickly and reliably apply this new technology to your SONET/SDH systems and implement a solution crafted to address your specific system requirements. For more information, visit [www.xilinx.com/products/design\\_resources/conn\\_central/index.htm](http://www.xilinx.com/products/design_resources/conn_central/index.htm). ❧