Serial FPDP

Serial FPDP - More Than an Extension of FPDP

Abstract

The Front Panel Data Port (FPDP) provides a high-speed connection between the A/D converter of a front-end data generating system (such as a radar antenna) and the digital signal processors (DSP) that are used to process the incoming data stream in real time.

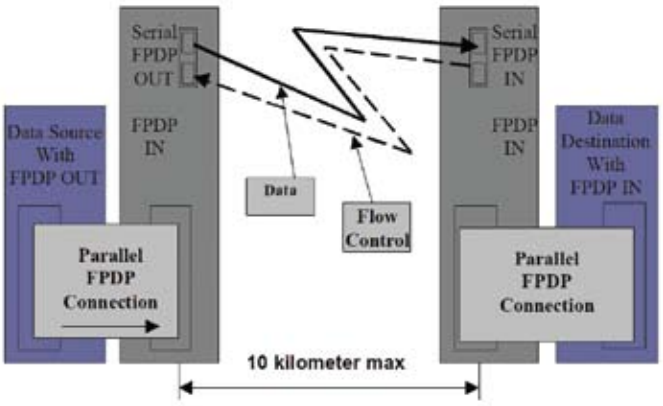
However, FPDP has a major limitation – the front-end data generation system must be located within one to five meters of FPDP cable length (depending on the FPDP configuration) of the DSP system. This presents a problem in many real world applications, that require greater distance between sensor and processor. Land-based radar systems, for example, may require that the A/D subsystem be located as close as possible to the antenna, and the processor system be located in an area quite distant from the antenna.

Serial FPDP (VITA 17.1), the technology pioneered in our FibreXtreme products, was developed to overcome this distance limitation. It presents a straightforward approach to serializing the FPDP data stream and transmitting it over extended distances. This paper outlines the concept of Serial FPDP (sFPDP), provides solutions for several types of real-time data scenarios, and explains supported topologies.

Introduction

SFPDP extends the maximum distance of FPDP connections by serializing the FPDP data stream and transmitting it over extended distance using copper or fiber optic cable. On the transmit side, the parallel FPDP data is converted to sFPDP. On the receive side, the reverse is true – the sFPDP data is converted back into the FPDP parallel data stream. The basic FPDP-to-sFPDP transaction process is also straightforward. The 32-bit FPDP input data passes through the transmit FIFO to the 8B/10B Encoder. The encoded data is then inserted into a continuous stream of idle characters being sent by the serializer, and the serialized stream is transmitted across the media (either fiber or copper) by the transceiver. Figure 1 illustrates this concept.

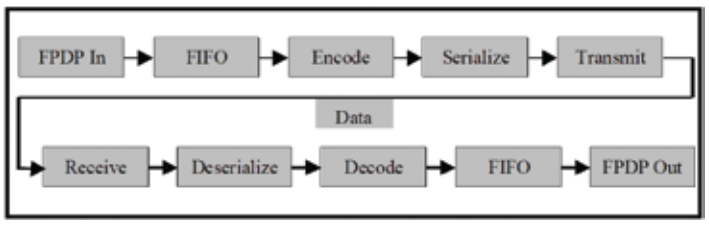
Figure 1: Serial FPDP Concept



On the receive end, the data is deserialized, decoded, and placed in the receive FIFO for sending via the standard FPDP connection. Figure 2 illustrates this concept.

CURTISS WRIGHT Controls Electronic Systems

Figure 2: FPDP to sFPDP Initial Process Flow



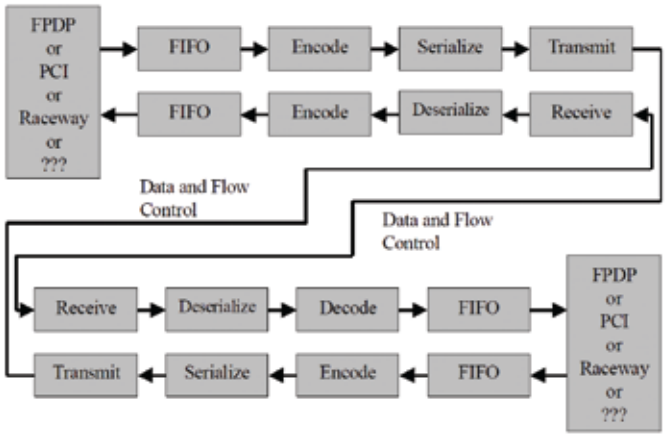
Expanding sFPDP

SFPDP has moved beyond this initial process flow and has become more than an extension of FPDP. It provides solutions for many types of real time data streaming scenarios – many of which are not FPDP applications. SFPDP has evolved into a "general-purpose" high-performance data link, providing high speed, bidirectional data flow (with flow control) across multiple platforms. It supports both point-to-point and broadcast topologies. It also has the flexibility to support other buses (PCI, RACEway) and has been implemented across multiple form factors (PCI, PMC, VME). Software communications drivers have been developed for a variety of operating systems. Figure 3 expands the initial sFPDP concept and provides a more representative view of this expanded sFPDP process flow.

SFPDP Protocol

In addition to transferring FPDP standard 32-bit data words, sFPDP also transmits the control signals used by FPDP. These signals are encoded as part of a series of special characters that are used to frame the data

Figure 3: Expanded sFPDP Processing Flow



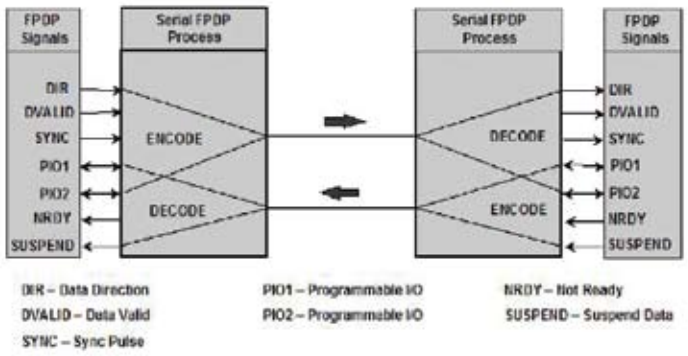
for transmission across the link. Three FPDP signals (DIR, DVALID, SYNC) are set by the FPDP (and sFPDP) transmit operation and two FPDP signals (NRDY, SUSPEND) are set by the receive operation.

Two other FPDP signals (PIO1 and PIO2) are user definable and are available to be set/read from either side of the link. Figure 4 shows the FPDP control signal flow across the Serial FPDP link. These signals are accessed directly from the input source when that source is actually FPDP. They are typically accessed from a special register that is updated either by custom hardware or the software driver in non FPDP applications.

SFPDP protocol uses the Fibre Channel link layer protocol to actually transmit the data. The standard Fibre Channel link layer protocol defines certain data sets in the 8B/10B encoding scheme as ordered sets. These ordered sets denote special control information for Fibre Channel. SFPDP uses a subset of these same ordered sets, but assigns them a different meaning. Using the same 8B/10B encoding as Fibre Channel allows the same components (ENDEC, SERDES) developed for Fibre Channel to be used to implement the sFPDP link layer.

There are eighteen ordered sets used by sFPDP to denote different information. Eight start-of-frame (SOF) sets are used to embed three bits at the start of a frame. The SOF ordered sets embed three FPDP signals – PIO1, PIO2, and DIR. Four status end-of- frame (SEOF) sets are used to embed two bits at the end of the frame. The four EOF ordered sets embed the FPDP signals NRDY and Overflow.

Figure 4: FPDP Control Signal Flow



There are two additional EOF ordered sets used by sFPDP to denote the actual end-of-frame. The Mark EOF (MEOF) denotes a frame that has SYNC associated with it, and the Frame EOF (FEOF) denotes a normal data frame. The other four ordered sets are inter-frame padding used to denote flow control informa tion and alternate frame interpretations.

SFPDP uses a series of frames to structure the data flow. These ordered set are used as part of this framing structure.

SFPDP has three basic frame types - a data frame, a SYNC without data frame, and a SYNC with data frame. The data per frame is variable, with a maximum size of the data buffer in a single frame of 512, 32-bit words (2048 bytes). SYNC is used to delimit data streams and maintain host program synchronization.

Figure 5: Serial FPDP Framing Protocol with Ordered Sets

Normal Data Frame
(Sent Empty when no data is available)

IDLE	SOF	Payload Between 0 and 512 32-bit words of data	CRC	FEOF	SEOF	GO/ STOP

Sync without Data Frame

IDLE	SOF	Payload Between 0 and 512 32-bit words of data	CRC	FEOF	SEOF	GO/ STOP

Sync with Data Frame

SWDV	SOF	Payload Between 0 and 512 32-bit words of data	CRC	FEOF	SEOF	GO/ STOP

ORDERED SETS
IDLE: An ordered set used to pad a DATA or SYNC without DATA Frame.
SWDV: An ordered set used to pad a SYNC with DATA Frame.
SOF: An ordered set used to pad and to indicate the start of data in frame. It carries the state of three signals PIO1, PIO2, and DIR.

OPTIONAL DATA (not an ordered set)
CRC: If disabled, CRC is not present.
If enabled and there is data in the frame, CRC is valid and checked.
If enabled and there is no data in the frame, CRC is invalid and ignored.

TYPE OF FRAME
FEOF: DATA Frame.
MEOF: SYNC Frame.
SEOF: Carries the state of two signals, NRDY and transmit overflow.

FLOW CONTROL FOR RETURN LINE
STOP: Suspend.
GO: OK to transmit.

Whenever a SYNC is recognized, the current frame is terminated and the proper SYNC frame (SYNC withdata or SYNC without data) is sent. Figure 5 shows the three types of frames and the ordered set placement within those frames.

Communications Link Efficiency

Real-time data streaming applications demand an effective communications scheme – one that efficiently utilizes the bandwidth of the link to move data, not to process overhead structure. Serial FPDP's point-to-point architecture meets this requirement by eliminating any overhead associated with arbitration and node addressing. The very minimal set of status and control signals (based on FPDP) eliminate the need for elaborate frame headers.

A sFPDP data frame consists of either five or six 32-bit data words (five without CRC and six with CRC). Assuming the allowable maximum data frame size of 2048 bytes or 512 words (32-bit), the actual data throughput on the link is in excess of 99% of the theoretical maximum. Based on the current standard link bandwidth of 1.0625 Gbps, the actual data throughput is as follows:

$$\frac{512 \text{ words}}{(512 \text{ words} + 5 \text{ overhead})} \times 106.25 \text{ MB/s} \approx 105 \text{ MB/s}$$

$$\frac{512 \text{ words}}{517 \text{ words}} = 99\ \%$$

The current standard being develop under the VITA (VITA 17.1) organization also provides for a link speed of 2.5 Gbps. This link bandwidth would provide a throughput as follows:

$$\frac{512 \text{ words}}{(512 \text{ words} + 5 \text{ overhead})} \times 250 \text{ MB/s} \approx 248 \text{ MB/s}$$

Sustained throughput rates are easily maintained in a straight hardware solution, such as the FPDP to sFPDP VME product provided by Curtiss-Wright Controls Electronic Systems; or the RACEway to Serial FPDP solution provided by Mercury Computer. The throughput constraints in these direct implementations are clearly definable. They include: (1) the bandwidth of the data source (FPDP, RACEway) on the transmit side, (2) the bandwidth of the data destination bus

on the receive side, (3) the bandwidth of the serial link, and (4) the speed of the translation process. Constraints 1 through 3 are established by existing standards. Constraint number 4 depends on the availability of components (FPGA's, SERDES, etc.) to support the translation process in real time. Commercial components are available to support the current and proposed sFPDP link speeds (1.0625 Gbps and 2.5 Gbps).

Extending sFPDP into a more general I/O solution requires providing support for many common processing platforms – typically PCI bus-based systems. These systems require I/O interface products in various form factors – including PCI, PMC and CPCI. Additionally, the I/O is controlled by and must interface with a variety of operating systems, each with its own intricacies. The actual performance, as measured by sustained data throughput, is dependent on many variables – including the specific PCI system design (processors, memory speeds, etc.); the I/O handling efficiency of the different operating systems; the design of the driver software for these operating systems, and the user's application software. Due to these many variables, there is no direct method to calculate the efficiency of these common processing systems. Testing has proven to be the only method to determine the actual throughput rates.

In order to verify the effectiveness of sFPDP across numerous platforms, using a variety of operating systems, and under varying conditions, engineers at Curtiss-Wright Controls Electronic Systems performed a series of throughput tests using a representative sample of systems. The sample included over 20 different platforms using Curtiss-Wright Controls' commercially available software drivers for 7 different operating systems (Windows NT®, VxWorks®, Linux™, Solaris™, IRIX®, LynxOS®, and Digital Unix®).

The ability of the host processor to set up the data transfer (on the transmit side) and to handle the incoming data (on the receive side) proved to be the real bottleneck in obtaining the high sustained throughput rates promised by sFPDP. In order to determine this actual effect, the throughput tests were done using a series of buffer sizes – the larger the buffer size, the less host processing required. These particular tests demonstrate that a buffer size larger that 256 kB is typically needed to approach the maximum throughput rates supported by sFPDP, and that processor inefficiencies become significant with small data buffers.

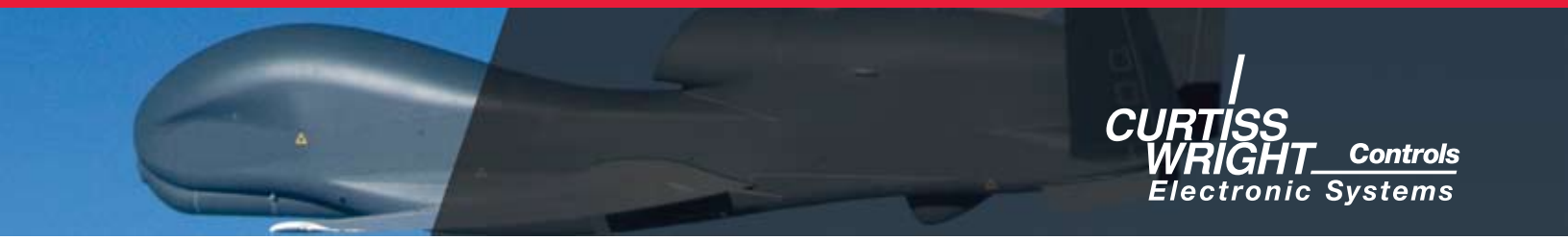
Although the results varied dramatically across the total test sample, sFPDP performed well among the most common PC and Power PC based systems when a reasonable buffer size was used. Serial FPDP proved itself able to operate above a 95% efficiency level (100 MB/s +) across a variety of platforms.

Supported Topologies

In addition to efficiency, a general-purpose data transfer protocol must support more than a simple point-to-point solution. One of the most commonly mentioned shortfalls of point-to-point protocols is inability to have one source and multiple destinations. A typical application using this feature would be one where data is sent to one system for processing and another system storage and archiving.

Serial FPDP supports a broad range of topologies – including Point-to-Point, Chained, Single Master Ring, and Multiple Master Ring.

- Point-to-Point topology is the native mode for sFPDP. It consists of a single transmitter and a single receiver. Data flows from transmitter to receiver.



- Chained topology consists of a single transmitter and a string of receivers. The data is received and passed on by each subsequent receiver, with this last receiver on the chain terminating the flow.

- Single Master Ring topology consists of a single transmitter and multiple receivers configured in a ring. The data is received and passed on by each subsequent receiver, with the last receiver on the loop sending the data back to the transmitting node.

- The Multiple Master Ring topology is another form of ring topology. It allows for multiple masters on the ring, but has some restrictions. All destinations physically located past a new master on the ring cannot receive the data from previous masters on the ring.

Summary

In summary, sFPDP has moved beyond being just an FPDP extender. It is a flexible and efficient protocol that is easily adapted to a variety of real time data collection and processing systems. It is a standard (VITA 17.1) that has been adopted for use in a wide range of applications – including radar systems, medical imaging systems, high resolution camera systems, and a variety of unique DSP applications.

Additional Information

Additional information about the sFPDP protocol, standards, working groups, and related issues can be obtained from:

The VMEbus International Trade Association
7825 E. Gelding Dr.
Ste. 104
Scottsdale, AZ 85260
Phone: (480) 951-8866
Email: info@vita.com
Web site: www.vita.com