DESIGN CONSIDERATIONS FOR ADAPTING LEGACY SYSTEM ARCHITECTURES TO SPACEWIRE

Session: SpaceWire Networks and Protocols

Short Paper

Robert Klar, Christopher Mangels, Sandra Dykes, and Michael Brysch

Southwest Research Institute, San Antonio, TX, 78238

E-mail: robert.klar@swri.org, christopher.mangels@swri.org, sandra.dykes@swri.org, michael.brysch@swri.org

ABSTRACT

Since first developed and standardized, SpaceWire has rapidly gained acceptance for use in space applications. SpaceWire offers many advantages to system designers over other on-board communications technologies. It is rather simple to implement, requiring relatively few logic gates and little interface memory to implement. Built upon low-voltage differential signaling (LVDS), it provides for high-speed communications, supporting link speeds well in excess of 100 Mbps in practical systems. It uses point-to-point links to connect nodes rather than a shared-bus architecture and thus provides much flexibility for incorporating redundancy into spacecraft systems.

Prior to SpaceWire, many spacecraft bus architectures were based on communications technologies such as the MIL-STD-1553B and RS-422 serial links. This paper describes a basic design process and suggests some considerations for adapting such legacy systems to use SpaceWire.

Initially, a system designer must analyze data flows to evaluate direction, volume, criticality, and timing requirements. For time-critical data, a synchronous schedule is often preferable and can be accommodated easily by making use of the time codes feature of SpaceWire. After critical data flows have been identified, redundancy and retransmission can be used to guarantee delivery of important data. Incorporating routing switches into the system can offer much flexibility for architecting robust redundant topologies. Additional reliability can be gained by suitably applying higher-level protocols such as Remote Memory Access Protocol (RMAP) to support data acknowledgement and retransmission. We present some practical considerations from project experience.

1 INTRODUCTION

Since it was standardized by the European Space Agency (ESA), SpaceWire [1] has been proposed for use on many space missions [2]. SpaceWire is a low-power, high-speed communications technology that was designed to be simple to implement in digital logic. It is rapidly replacing widely prolific legacy communication technologies such as RS-422 and MIL-STD-1553B. Since designs are often reused across several missions to reduce development costs and schedule, it is important to consider the characteristics and advantages of these legacy systems when moving towards a SpaceWire architecture. This paper describes a basic design process and suggests some considerations for adapting such systems to use SpaceWire.

2 LEGACY ONBOARD COMMUNICATIONS SYSTEMS

Originally published by the United States Department of Defense as “Military Standard Aircraft Internal Time Division Command/Response Multiplex Data Bus” [3], MIL-STD-1553B, as the name reveals, was initially developed for application in military aircraft. However, over time, it has found numerous other applications. The United States National Aeronautics and Space Administration (NASA) has found it particularly useful because of its durability and suitability for high-radiation environments. It has been widely adopted for use in command and data handling systems on numerous flying spacecraft including the recent Swift and Fermi (i.e. formerly GLAST) missions [4,5].

MIL-STD-1553B has endeared itself to system designers for many reasons. Since it is based on a shared bus topology, devices must share the transmission medium by time-division multiplexing (TDM), staggering transmission of data on the bus in time. This characteristic of MIL-STD-1553B results in a deterministic real-time schedule for data collection (i.e. the bus schedule).   Spacecraft control loops are often designed around this bus schedule. In the terminology of the MIL-STD-1553B standard, the bus schedule is managed by a Bus Controller (BC). A device on the bus with which the BC communicates is a Remote Terminal (RT). MIL-STD-1553B supports half-duplex communication in which the RT responds only to commands directed to it by the BC. Although BC implementation is not defined by the standard, most modern systems use a frame controller which can process a sequence of command messages in a repetitive fashion [6].

A beneficial feature of MIL-STD-1553B is its rich support for fault-tolerance. Developed for use on military aircraft, it was designed with support for redundancy. A common configuration consists of a primary bus and a single hot spare [6]. Although the standard provides for higher levels of redundancy, these are rarely seen in modern systems because it results in reduced bus throughput. Since messages have specific timing requirements and include parity, BCs can detect errors in transmissions. To handle errors, BCs often perform one or more retries on a bus before autonomously switching over to a redundant bus.

Although well suited for command and control applications, MIL-STD-1553B is not well suited for bulk data transfers [13]. With a bus clock rate of 1 MHz, the data rate is limited to only a few hundred kilobits per second in practical use. Consequently, many spacecraft subsystems and instruments producing large quantities of data instead use a dedicated high-speed interface. Because they are relatively simple to implement, serial data interfaces are common. Until more recently, with advent of Low-Voltage Differential Signaling (LVDS), RS-422 signaling was often used to implement serial interfaces in spacecraft systems.

RS-422 commonly refers to ANSI/TIA/EIA-422-B, “Electrical Characteristics of Balanced Voltage Differential Interface Circuits” [7], or the equivalent international standard, ITU-T Recommendation V.11 [8]. As the name reveals, characteristics of RS-422 include a differential driver and receiver. RS-422 supports relatively high data rates, with practical implementations of up to 10 Mbps.

For low-speed data rates up to 115,200 baud, RS-422 is often used to implement an asynchronous serial interface. There is a variety of off-the-shelf hardware to support asynchronous serial communications with a relatively inexpensive personal computer (PC). For higher data rates, a synchronous serial interface is more common. A unidirectional configuration for transferring telemetry is often employed in spacecraft systems and might typically consist of three signal pairs: clock, frame, and data.

3 COMMUNICATIONS SYSTEMS DESIGN

When designing an onboard communications system, it is often preferable to take a top-down approach to solving the problem. A basic design process could include the following steps: 1) Identify top-level requirements; 2) Determine link rates; and 3) Design and size each link.

The first step in the process is to identify top-level requirements. One helpful technique for deriving these requirements is to construct a mission-level data-flow diagram. The diagram can then be analyzed to identify producers and consumers of data. A key part of this analysis is to identify the quantity of data that needs to be transferred on board.

The next step is to determine the required link rates. For some data producers, data rates are easily computed by reviewing the sampling rate and the number of bits in each sample. However, for others, determining data rates can be more difficult. Event detectors, for example, often produce burst data which may be difficult to characterize. In this case, an accurate data model is sometimes needed to estimate the worst case.

After data rates have been determined, the final step is to design and size each link appropriately. Although legacy interfaces may still be viable, SpaceWire can provide similar benefits and is well suited to replace them for many applications. Many vendors produce SpaceWire products and rich variety of offerings is now readily available on the market.

4 SYNCHRONIZED AND UNSYNCHRONIZED ARCHITECTURES

SpaceWire offers significant flexibility to communications systems designers in that it supports both synchronized and unsynchronized architectures. In designing a communication system, it is important to understand the specific nature of the data being produced. If the data is produced at a regular cadence, then a time-synchronized or spacecraft spin-synchronized data system might work well. On the other hand, if the data is produced in bursts or at irregular intervals, then an unsynchronized data system would likely be a better choice. Hybrid architectures are certainly possible and well supported by SpaceWire.

SpaceWire includes the capability to send a low-latency signal called a time code. In practical systems, this can be used for synchronization to an accuracy of a few microseconds without enhancements [9]. Time codes can be used to frame data into discrete time slots to implement time-division multiplexing similar to that inherently provided by MIL-STD-1553B. The time code feature has been further extended to provide even more utility. Researchers at NASA Goddard Space Flight Center (GSFC) developed an intellectual property (IP) core that provides up to 4 unique time codes that can be used to communicate side-band data as well as time [10]. Researchers at St. Petersburg University have developed a scheme for using time codes as distributed interrupts for support of hard real-time control systems; the protocol provides for up to 16 possible unique codes [11].

High-speed serial interfaces have often been employed to provide unsynchronized telemetry to onboard data storage systems. Significant engineering time and effort can be spent developing and specifying unambiguous, detailed requirements for the required data framing. Consequently, many variations in methods for determining these requirements exist. Substituting SpaceWire for a high-speed serial interface can greatly simplify the interface definition task, as it provides a standard mechanism for defining packet boundaries.

5 IMPROVING RELIABILITY

Proper application of redundancy can improve reliability. SpaceWire provides many options for incorporating redundancy. Several are shown in Figure 1. One option is to include a router in the host system. NASA has developed a router IP core with 4 external ports and 4 internal ports that is sized to fit within a single Actel RTAX2000 with margin. This structure provides an extremely versatile building block that can be used to construct a variety of redundant topologies. For small systems, another option is to simply include redundant links. Link cores require few logic gates and are easily accommodated by modern space-qualified programmable devices. Researchers at Southwest Research Institute ® (SwRI®) developed a link core that consumes less than ten percent of an Actel RTAX2000. A third option is to use a redundant physical layer. Researchers at NASA GSFC have developed a link-and-switch implementation for SpaceWire that allows for an automatic switchover to a redundant link in the event of a link failure. In some ways, this is analogous to the automatic bus switching provided by MIL-STD-1553B.

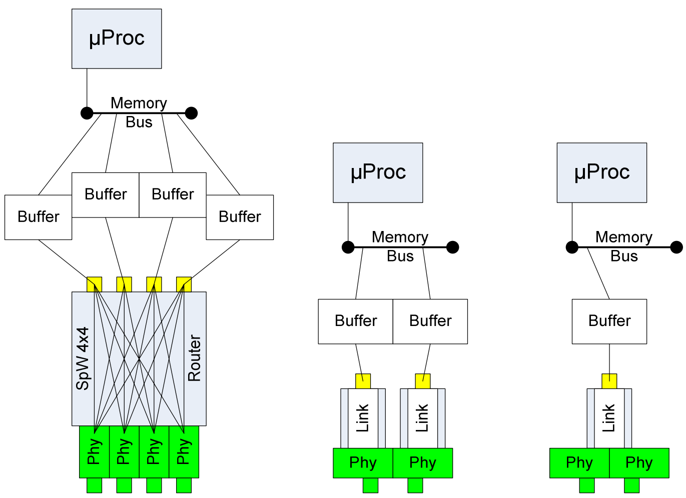

Figure 1. SpaceWire Redundancy Options

The SpaceWire Working Group (SWG) created a specification for a standardized SpaceWire encapsulation header to provide a means to multiplex and demultiplex packets of different higher-level protocols [12]. Included in this standard is the definition of the Remote Memory Access Protocol (RMAP). RMAP supports a command/response mechanism similar in many respects to the operation of MIL-STD-1553B. It provides mechanisms for acknowledged and verified data transfers. RMAP has further advantages in that it incorporates strong error-detection by inclusion of a cyclic-redundancy code (CRC). When RMAP is used in combination with redundant links, it can emulate the operation of MIL-STD-1553B with redundant busses.

6 Conclusion

We have introduced some important considerations for a system designer wanting to use SpaceWire. SpaceWire can be a versatile and robust solution for constructing an onboard communications system.

7 References

1. ECSS-E-ST-50-12C, Space Engineering: SpaceWire – Links, nodes, routers, and networks", ESA-ESTEC, July 2008.
2. ESA SpaceWire Website:
   http://spacewire.esa.int/content/Missions/ESA.php
   http://spacewire.esa.int/content/Missions/NASA.php
   http://spacewire.esa.int/content/Missions/JAXA.php
3. "Military Standard Aircraft Internal Time Division Command/Response Multiplex Data Bus," MIL-STD-1553b, United States Department of Defense, 21 September 1978.
4. "Swift 1553 Bus Protocol Interface Control Document." Spectrum Astro Inc., August 2000.
5. "Gamma-Ray Large Area Space Telescope (GLAST) 1553 Bus Protocol Interface Control Document." Spectrum Astro Inc., December 2002.
6. "A MIL-STD-1553B Tutorial," Document #1600100-0028, Condor Engineering, June 2000.
7. "Electrical Characteristics of Balanced Voltage Digital Interface Circuits", TIA/EIA-422-B, May 1994.
8. "Electrical characteristics for balanced double-current interchange circuits operating at data signalling rates up to 10 Mbit/s", ITU-T Recommendation V.11, October 1996.
9. Barry Cook. "Time-code enhancements for SpaceWire", MAPLD International Conference, 2006.
10. NASA GSFC Innovative Partnerships Program Website:
    http://ipp.gsfc.nasa.gov/ft-tech-spacewire.html
11. Yuriy Sheynin, Sergey Gorbatchev, Ludmila Onischenko. "Real-time Signaling in SpaceWire Networks. International SpaceWire Conference, 2007.
12. ECSS-E-ST-50-11C, Space Engineering: SpaceWire protocols", ESA-ESTEC, July 2008.
13. Ronnie Killough. "Integrating CCSDS and MIL-STD-1553B: What You Should Know." IEEE Aerospace Conference, December 2001.