

DESIGN AND IMPLEMENTATION OF AN AVIONICS FULL DUPLEX ETHERNET (A664) DATA ACQUISITION SYSTEM

Alberto Perez, Technical Manager, System Test & Integration
John Hildin, Director of Network Systems
John Roach, Vice President of Network Products Division
Teletronics Technology Corporation
Newtown, PA USA

ABSTRACT

ARINC 664 presents the designers of data acquisition systems challenges not previously seen on other aircraft avionic buses. Among the biggest challenges are providing the test instrumentation system with the capacity to process two redundant Ethernet segments that may be carrying packet traffic at near wire-line speed. To achieve this level of performance, the hardware and software must not only perform mundane operations, like time stamping and simple virtual link MAC filtering, but also need to implement core ARINC 664 functions like redundancy management and integrity checking. Furthermore, other TCP/IP operations, such as IP header checksum, must also be offloaded to the hardware in order to maintain real-time operation. This paper describes the implementation path followed by TTC during its development of an ARINC 664 network monitor used in a large commercial aircraft flight test program.

KEY WORDS

AFDX, ARINC 664, Ethernet, flight test, IP, redundancy

INTRODUCTION

ARINC 664 Part 7 (A664) is a standard that defines the electrical and protocol specifications for a communications network suitable for use in an avionics environment. This standard is based on earlier work done by Airbus called AFDX (Avionics Full Duplex Switched Ethernet) and used for the A380. The main objective of ARINC 664 is the creation of a deterministic data network that can be used for flight critical applications. This is accomplished by providing dedicated bandwidth to each communication path in the network and allowing for the specification of the quality of service (QOS) available to each node in the system.

An ARINC 664 network consists of aircraft host computers and switches. In the ARINC 664 network, the aircraft computers are the consumers of communication services in the network. The End System (ES) layer, within the host computer, provides this service. The End System enables the software applications running on the host computer to send and receive data in a reliable and secure way. Thus, the End System not only corresponds to the concept defined in the OSI protocol suite [1], but also has an additional layer that gives the ARINC 664 network its deterministic characteristics.

End Systems are interconnected using packet switching computers called “AFDX switches”. The switches’ main task is to exchange frames between the End Systems connected to its ports. In addition, the switches enforce traffic policies on the preconfigured virtual links that the users statically define. A Virtual Link (VL) is a pipe employed by an End System to transmit data across the network. A Virtual Link is a unidirectional point to multipoint communication path that provides service analogous to those found in Asynchronous Transfer Mode (ATM) Constant Bit Rate (CBR).

| VL | Bandwidth Allocation Gap (BAG) in msecs | Maximum Frame Size (Lmax) in bytes | Vehicle System | Identifier |
|----------|---|------------------------------------|----------------|---------------------------------|
| 1- - - - | 2 | 420 | MCS | Main Computer System |
| 2- - - | 4 | 64 | BC | Brake Controller |
| 3- - - - | 8 | 1460 | GPS | Global Position System |
| | | | MnDAU | Miniature Data Acquisition Unit |

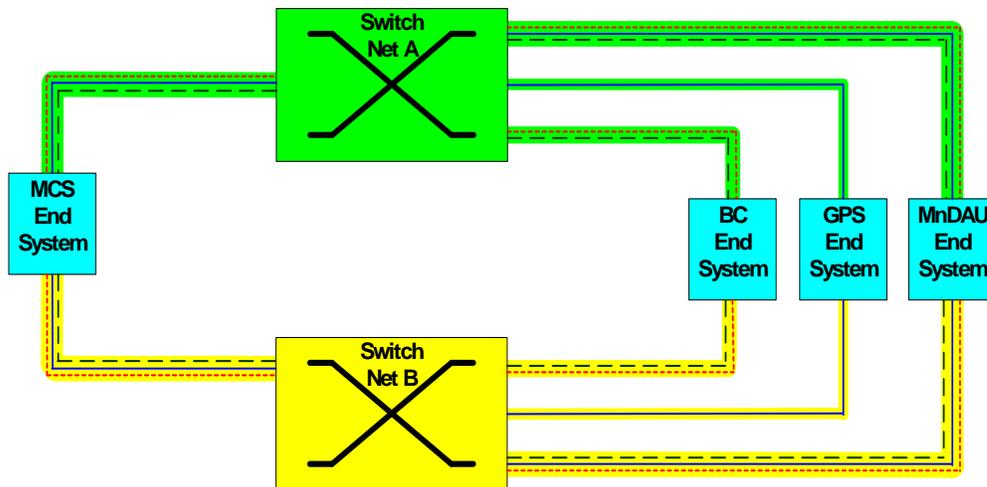


Figure 1: Simple AFDX network

Figure 1 demonstrates two important concepts of an ARINC 664 network.

1. Redundancy

This property of the ARINC 664 network is accomplished by having two independent paths between each End System. For each Virtual Link, a transmitting End System sends a copy of the data to both networks. The receiving End System accepts the first instance of the data to

arrive and silently discards the other copy. In Figure 1, the MCS End System is sourcing all three virtual links on the network. The system designer pre-configures the VLs for redundancy, forcing the End System to transmit the same frame on both Network A and Network B.

2. Traffic Shaping

Each Virtual Link regulates its maximum transmit rate by using two configurable parameters:

- a. *Bandwidth Allocation Gap* (BAG) that regulates how often the Virtual Link is allowed to transmit on the network and
- b. The *Maximum Frame Size* (Lmax).

Computing the bandwidth allocated to a Virtual Link is a function of the BAG in Hz and Lmax in bytes. For example, in our simple A664 network, the maximum bandwidth allocated to virtual link 1 is computed by multiplying Lmax by the rate in seconds ($420 * 1000/2$) = 210Kbytes/sec.

These two properties make the ARINC 664 network very predictable in terms of the rate of traffic that an End System is expected to initiate or terminate. The A664 network system designer knows ahead of time the resources that the network is going to need and programs the AFDX switches accordingly. The engineer that designs an aircraft End System selects a processor capable of handling the traffic load that the End System is expected to process and does not need to worry about any other traffic patterns on the network since the switch and the built-in MAC filters will discard all the traffic that does not belong to the End System. The engineer that designs the Fight Test Instrumentation (FTI) system that captures the network traffic has no previous knowledge of what is expected of the FTI End System. All the FTI engineer knows is that the system must be able to handle two full-duplex 100Mbps lines.

Consider the following instrumentation requirements:

- a. Terminate up to 65535 Virtual Links
- b. Must be able to process up to 148,800 frames per link
- c. Selectively perform Redundancy Management and Integrity Checking
- d. Keep statistics for each Virtual Link
- e. Timestamp all received frames within 100 nanoseconds

The approach chosen by Teletronics Technology Corp (TTC) was not only to offload traditional functions such as timestamping to the hardware, but also the Virtual Link protocol defined in ARINC 664 Part 7[2].

IMPLEMENTATION

The Mn664-2000L is a miniature networked encoding unit that processes and delivers packetized A664 data to designated nodes in the network. It is designed to capture packets received from ARINC-664 switches and feed processed frames into the network. It consists of a unique

ARINC-664 interface module using a custom A664 FPGA, an IEEE 1588 time and Ethernet interface module, a processor module running real-time Linux, and a 15-watt power supply. See Figure 2 for an instance of the Mn664-2000L components.



Figure 2: Mn664-2000L

The hardware platform chosen in the design of the A664 MnDAU combines the flexibility of two processors running real time Linux with the performance of two dedicated FPGAs. One processor is dedicated to servicing the A664 protocol while the second performs data encapsulation and transmission to the data acquisition network. The actual breakdown of functionality in the unit is as follows:

Table 1: Mn664-2000L Functional Partitions

| Module | Description |
|-------------------------|---|
| ARINC-664 bus interface | <ul style="list-style-type: none"> ■ I/O board: one A664 channel pair (Tx/Rx * 2), an A664 FPGA, and bidirectional optical transceivers ■ Slave processor board: <ul style="list-style-type: none"> ▣ 32MB RAM, 64MB Flash memory ▣ Memory bandwidth 533 MBps peak ▣ Integrity checking and redundancy management compliant with ARINC-664 Part 7 ▣ Timestamps and DMAs the captured ARINC-664 packets ▣ Serial RS-232 engineering port |

| Module | Description |
|---------------------------------------|--|
| Main processor | <ul style="list-style-type: none"> ▪ Onboard 32MB RAM and 64MB Flash memory ▪ Serial RS-232 engineering port ▪ 1PPS signal output ▪ Software-based real-time Linux ▪ ARINC-664 and network driver |
| IEEE 1588 time and Ethernet interface | <ul style="list-style-type: none"> ▪ IEEE 1588 time synchronization ▪ 10/100 Base-T Ethernet port |
| Power supply | <ul style="list-style-type: none"> ▪ 28 VDC +/-4V with input filter ▪ 15 Watts output |

Figure 3 below shows two processors that are connected via a PCI Bus in a master-slave configuration. The processor directly connected to the FPGA is the slave processor. This CPU interfaces to the A664 FPGA through the processor's local bus. The function of the master processor is to program and control the acquisition of frames within the slave processor.

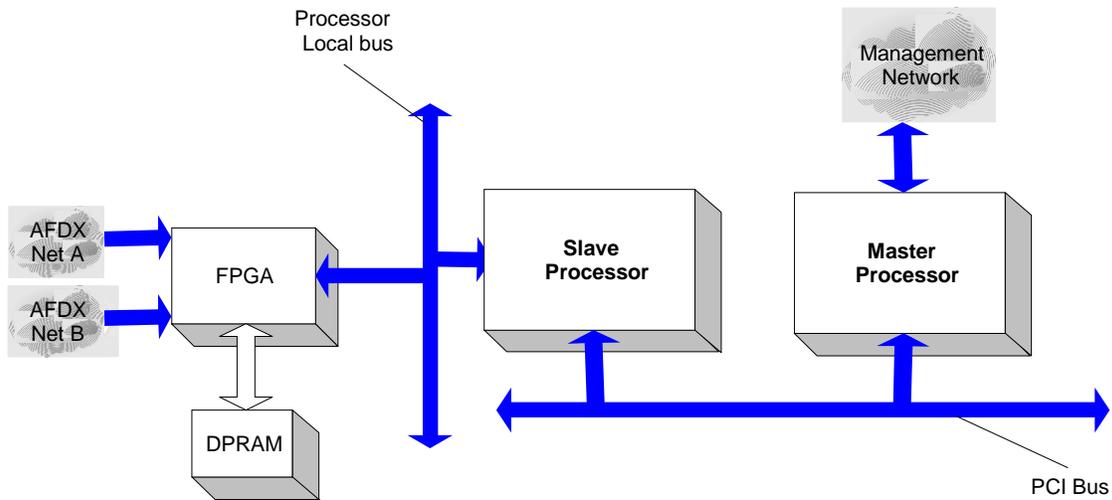


Figure 3: A664 MnDAU Platform

A664 FPGA DEVELOPMENT

During the development of the ARINC 664 MnDAU, the main focus was on the design of the FPGA that was handling the redundant frames. To support fast and efficient processing of frames, the FPGA has a local dual-port RAM. It is used as a direct lookup of Virtual Link information and to buffer incoming frames until the FPGA is ready to handle the received data. This FPGA is placed on the MII bus between the PHYs and the slave PowerPC processor. This means that the FPGA has to compute the frame check sequence (FCS-32) on all incoming frames as well as generating the FCS-32 once the A664 sequence number is extracted and the timestamp is appended to the frame.

To leverage the hardware, all the Virtual Link functions defined in the ARINC 664 specification part 7, filtering and statistic collection were implemented in the FPGA.

These functions include the following:

- ARINC 664 functions
 - Integrity checking per Virtual Link ID and network
 - Redundancy management per Virtual Link ID, SkewMax and sequence number range
- Filters
 - Defaults to discard all incoming A664 frames
 - Filters based on Virtual Link ID
 - Filter on source and destination IP address
 - Filter on source and destination UDP port
 - Integrity based on virtual link ID
 - Redundancy based on virtual link ID
- Ethernet
 - Computes the receive frame check sequence
 - Discards A664 frames with errors
 - Provides the status of:
 - Bad CRC
 - Frame errors including run, overflow, byte and non-UDP
 - Integrity resets
 - Replaces the source MAC address and type with the eight bytes of timestamp
 - Recomputes the CRC-32
 - Retransmits to the MAC devices

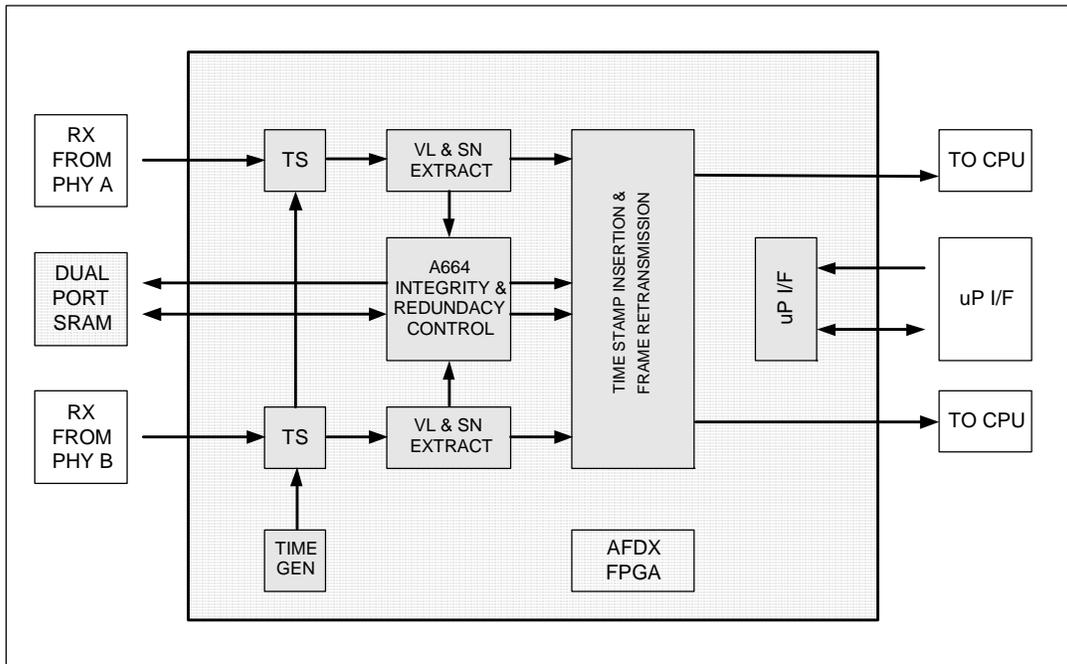


Figure 4: FPGA Architecture

SLAVE PROCESSOR

The Slave processor core function is to program the Virtual Link table stored in the FPGA's DPRAM and to receive the frames that passed the Integrity Checking and Redundancy Management procedures. This processor provides 32 Mbytes of DRAM that is used for program, frame storage and IP reassembly queues. Most of the work of the Slave processor is in setting up data transfers across the PCI bus. For this task, embedded DMA engines within the processor are used to perform the actual data transfer. The use of DMA makes the end-to-end delivery of data without the need of processor driven I/O. Another critical task performed by the Slave processor is the reassembling of fragmented IP packets. The IP fragmentation and reassembly procedure is described in IETF RFC 791[3].

MASTER PROCESSOR

The master processor for the Mn664-2000L stores the program code and configuration data used by the system. The master processor executes the operating system, drivers, and application software. All user interactions with the system occur through the master processor. In addition, the module containing the master processor provides the connectors that attach to other modules in the Mn664-2000L. Each connector has a combination of buses, clocks, and voltage rails. The following buses are supported between the master processor module and other modules:

- The master processor local bus is connected to the module which implements the data acquisition Ethernet interface
- The master processor MII Bus is connected to the IEEE 1588 module that implements the PTP protocol
- The master processor PCI Bus is connected to the slave processor module
- A backplane connector is used to distribute power to all the modules in the stack

CONCLUSION

Specialized avionics communications buses require specialized instrumentation systems to interface and record data from them. The Mn664-2000L is an example of a highly specialized device that can accept data from two fully redundant Ethernet interfaces running at near line rate speeds and still meets the requirements commonly expected of instrumentation test equipment. This paper has provided an outline of the approach taken by one company to draw a practical line between software and hardware implementation constraints that provided a cost effective and efficient solution to a difficult testing problem.

REFERENCES

- [1] End Systems to Intermediate System Routing Exchange Protocol, ANSI X3S3.3, published as RFC-995, April 1986.
- [2] ARINC Specification 664 Part 7, AEEC, September 2004.
- [3] Postel, J., "Internet Protocol [IP]," RFC-791, September 1981.