

Developing AFDX Solutions

Introduction

As the complexity of avionics systems has grown, for both flight-critical items and passenger entertainment, so has the need for increased bandwidth of on-board data buses. The desire for rapid deployment with minimal development and implementation costs, such as wiring, has driven the industry to explore existing off-the-shelf technologies.

Both Boeing and Airbus have explored commercial Ethernet technology to build a next-generation avionics data bus. This research has resulted in the development of Avionics Full-Duplex Switched Ethernet (AFDX), based upon IEEE 803.2 Ethernet technology, but adding specific functionality to provide a deterministic network with guaranteed service. Currently, Aeronautical Radio, Inc. (ARINC) is developing a standard based upon Ethernet technology: ARINC 664.

This application note begins with an overview of AFDX and ARINC 664. Following the overview is an explanation of how engineers could implement an AFDX-compliant interface (ARINC 664, Part 7) using Actel devices and intellectual property. Core10/100, an Ethernet MAC, and Actel FPGAs can be used to develop AFDX End Systems (ESes) for use in endpoints and switches. This application note describes a potential prototyping solution using the Platform8051 Development Kit and concludes with a summary of the specific advantages Actel devices provide in avionics applications.

Currently, the Actel solution for AFDX includes Core10/100, the PF8051 Development Kit, and this application note.

Commercial Aircraft Data Networks

Before considering the design of an AFDX interface, this section will review the recent history of aircraft data networks (ADNs).

ARINC 429

The most widely used commercial aircraft data network in existence today is ARINC 429. Developed over 30 years ago, it has proven to be highly reliable and can be found in a wide range of aircraft, including the McDonald Douglas MD-11, the Boeing 737, 747, 757, and 767, and the Airbus 330 and 340.

ARINC 429 is based upon a unidirectional bus with a single transmitter and up to 20 receivers. Data is communicated over twisted pair using Bipolar Return-to-Zero Modulation (BPRZ). The standard specifies a fixed data word of 32 bits that can be communicated over the bus at either high speed (100 kbps) or low speed (12.5 kbps).

Although ARINC 429 is referred to as a bus, its single transmitter architecture results in a point-to-point solution, requiring a large amount of wiring in order to connect various aircraft systems together. Despite its robustness and wide acceptance, the low data rates and the massive amount of wiring needed to connect systems have driven the need for more modern solutions. Actel provides an ARINC 429 solution that includes Core429 and the Core429-DEV-KIT.

ARINC 629

For the 777 project, Boeing developed ARINC 629, a higher-speed data networking solution, rather than continuing with ARINC 429. ARINC 629 is a bidirectional bus supporting data rates up to 2 Mbps with a maximum of 120 data terminals. The bus operates without the need for a bus controller—each terminal is self-monitoring, enhancing the reliability of the architecture.

Although ARINC 629 reduced the amount and weight of wiring needed in an aircraft and increased the data rate over ARINC 429 by 20 fold, it requires custom hardware, which adds to the overall cost of the aircraft. In addition, ARINC 629 has not met with acceptance by other aircraft makers, increasing its costs.

ARINC 664

In the late 1990s, work started to define a next-generation aircraft data network (ADN) based upon IEEE 802.3 Ethernet. The goal was to take advantage of commercial off-the-shelf (COTS) hardware to reduce cost and development time, while ensuring compatibility with the need for robustness and higher data rates in commercial aviation. Generally, ARINC 664 defines a profiled IEEE 802.3 network that uses IP addressing and transport protocols (TCP and UDP).

ARINC 664, currently undergoing adoption, defines a set of best practices and requirements for any next-generation ADN. The specification is broken into several parts:

- Part 1 – System Concepts and Overview
- Part 2 – Ethernet Physical and Data Link Layer Specifications
- Part 3 – Internet-Based Protocols and Services
- Part 4 – Internet-Based Address Structures and Assigned Numbers
- Part 5 – Network Interconnection Services and Interconnection
- Part 6 – Reserved
- Part 7 – Avionics Full Duplex Switched Ethernet (AFDX) Network
- Part 8 – Upper-Layer and User Services

Part 7 of the ARINC 664 specification defines a special application for a full-duplex, deterministic aircraft data network referred to as AFDX, the focus of this document.

AFDX

Airbus Industries began work to define a robust next-generation ADN for use in the A380. Avionics Full Duplex Switched Ethernet (AFDX) is a special application of a network compliant with ARINC 664. Though 802.3 Ethernet offers high speed and low cost due to widespread commercial usage, it does not offer the robustness required for an avionics system. The primary drawback of IEEE 802.3 is the lack of guaranteed bandwidth and Quality of Service (QoS). AFDX attempts to solve these issues while allowing the use of as much commercial 802.3 hardware as possible.

AFDX provides for a star topology of up to 24 ESes tied to a switch that can be bridged (cascaded) to other switches in the network. This deterministic network allows for link redundancy (dual physical links), guaranteed bandwidth, and QoS.

Part 7 allows for the mapping of other bus standards (e.g., ARINC 429 or MIL-STD-1553) onto the network and allows for communication with other ARINC-664-compliant but nondeterministic networks through gateways and routers.

Overview of AFDX

AFDX addresses the shortcomings of IEEE 802.3 Ethernet by adopting concepts from the telecom standard, Asynchronous Transfer Mode (ATM). These extensions to standard Ethernet make possible a deterministic network with guaranteed bandwidth and QoS.

The major aspects of AFDX are as follows:

- Profiled network – parameters for various ESes are defined in configuration tables loaded into the switch at start-up.
- Full duplex – the physical interconnect medium is twisted pair, with separate pairs for transmit and receive channels.
- Switched network – the network is wired with a star topology. Each switch connects a maximum of 24 ESes. Switches can be cascaded to construct a larger network.

- Deterministic – the network emulates a deterministic, point-to-point network through the use of virtual links and guaranteed bandwidth.
- Redundancy – dual networks provide a higher degree of reliability than a single network scheme provides.
- Performance – the network operates at either 10 Mbps or 100 Mbps. The default mode is 100 Mbps operation.

Virtual Links

At the heart of an AFDX network is the virtual link (VL). Each VL builds a unidirectional logic path from the one and only source ES to one or more destination ESes. Each VL is allocated dedicated bandwidth, the amount being defined by the system integrator. The total bandwidth of all created VLs cannot exceed the maximum available bandwidth of the network, and the bandwidth allocated to a given VL is reserved to that link.

To accommodate less critical data communication needs, AFDX also allows for the construction of sub-virtual links (sub-VLs). The data queues of all sub-VLs assigned to a single VL are read in a round robin sequence among the sub-VLs with data to transmit. Data from a single sub-VL cannot be split across multiple VLs. Although the bandwidth and latency of a VL are guaranteed, there is no guarantee for the sub-VLs. Note that from the Media Access Controller (MAC) layer down, a VL composed of multiple sub-VLs is handled the same way a single-data-stream VL is handled. The specification states that a VL queue is required to have the ability to handle four sub-VL queues at a minimum.

The number of VLs a single switch can support is limited to 4,096. In a cascaded switch network, the total number of VLs in the system will also be limited by any links that cross switch boundaries. The specification does not specify a limit for the number of VLs a single ES can handle. However, based upon the maximum bandwidth allocation gap (BAG) value of 128 ms and the minimum of 1 ms, the total number of VLs through an ES is limited to 128 (see the "[BAG](#)" section for more details). The specification does limit the number of sub-VLs that may be created within a single VL to four.

Data packets communicated across a VL must be sent in order of reception from the application (ordinal integrity). The "[Hardware Implementation](#)" section on page 6 explains how ordinal integrity is handled.

Guaranteed Service

Fundamental to an AFDX network is guaranteed service: both the bandwidth and maximum end-to-end latency of the link are guaranteed. However, there is no guarantee of packet delivery. Transmission acknowledgements and retransmission requests must be handled at the application level.

BAG

The primary bandwidth control mechanism is the BAG ([Figure 1](#)). The BAG defines the minimum time interval between the starting bits of two successive AFDX frames, assuming zero jitter. BAG values range from 1 ms to 128 ms (values must be a power of 2: $BAG = 2^k$, where k is an integer).

The BAG value, and thus the allocated bandwidth, for a given VL is determined by the system integrator as required by the application and equipment. BAG values for all VLs are stored in the appropriate ES and switch configuration tables.

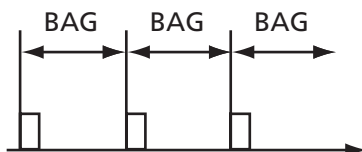


Figure 1 • Bandwidth Allocation Gap

Jitter

The ES may introduce jitter when transmitting frames for a given VL. This jitter is defined as the interval from the beginning of the BAG to the first sent bit of the frame being transmitted at the VL's maximum allocated bandwidth (Figure 2). This jitter may be introduced by the transmitting technology and the traffic-shaping function. A given ES may have to transmit data for multiple VLs, so a frame from one VL can be delayed up to the maximum allowed jitter value to limit the instantaneous ES frame rate and thus accommodate frames from other VLs.

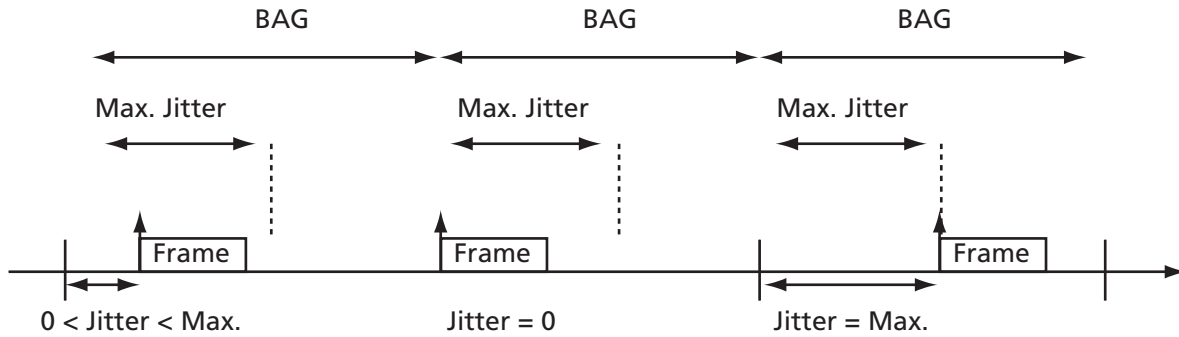


Figure 2 • Jitter Defined

The maximum allowed jitter for a given ES is defined in EQ 1.

$$\text{Max. Jitter} \leq 40 \mu\text{s} + \frac{\sum_{i \in \{\text{set of VLs}\}} (20 \text{ bytes} + L_{\text{MAX}}) \times 8 \text{ bits/bytes}}{N_{\text{BW}}}$$

EQ 1

where:

Max. Jitter is in μs .

N_{BW} is the medium bandwidth in bps.

L_{MAX} is the maximum allowed frame size for the VL in bytes.

An example for an ES with two VLs, operating at an average bandwidth of 50 Mbps is as follows:

For VL₁, $L_{\text{MAX}} = 1518$ bytes (maximum value)

For VL₂, $L_{\text{MAX}} = 64$ bytes (minimum value)

$$\begin{aligned} \text{Max. Jitter} &= 40 \mu\text{s} + [(1518 + 20 + 64 + 20) \times 8 \text{ bits/bytes}] / 50 \text{ Mbps} \\ &= 40 \mu\text{s} + 259 \mu\text{s} \\ &= 299 \mu\text{s} \end{aligned}$$

The specification allows 40 μs for the maximum technological jitter, and in no case shall the total jitter be allowed to exceed 500 μs .

Latency

Although ARINC 664 Part 7 does not specify a maximum system latency, any supplier is required to specify the upper limit of latency for any system delivered. The specification does set down limits for some aspects of system latency. These limits will be covered in detail in the design sections that follow.

Frame Format

The AFDX frame format is shown in Figure 3. The destination and source addresses listed contain the MAC addresses for the ESes. Actual IP address information is contained in the IP Structure block. The UDP structure identifies the appropriate application port. The AFDX payload ranges from 1 to 1471 bytes. Payload sizes less than 17 bytes must be padded to maintain a minimum length of 17 bytes.

7 bytes	1 byte	6 bytes	6 bytes	2 bytes	20 bytes	8 bytes	17 to 1471 bytes	1 byte	4 bytes	12 bytes
Preamble	Start Frame Delimiter	Destination Address	Source Address	Type IPv4	IP Structure	UDP Structure	AFDX Payload	Seq Number	Frame Check Seq	Interframe Gap

Figure 3 • AFDX Frame

The one-byte sequence number is used to maintain ordinal integrity within a given VL. The frame sequence number is initially set to 0 upon ES start-up or reset. During continuous operation, the number wraps back to 1 after reaching a value of 255.

The maximum frame size is set for each VL and is represented by the parameter L_{MAX} . The range of this parameter is between 64 and 1518 bytes.

Addressing

AFDX network addressing is based upon the MAC address of each ES. ARINC 664 does not specify an algorithm for assigning MAC addresses. This task is left to the system administrator and must be compliant with the IEEE 802.3 specification for locally administered addresses.

The MAC address is 16 bits in length. In Boeing applications, the full 16 bits are available for assignment. In Airbus applications, only the least significant 12 bits are used, and the most significant four bits are set to zero.

The source address must be a unicast address and follow the format detailed in the specification. The source address includes bits for identifying to which of the two redundant networks the MAC is attached. The destination address is a multicast address that includes a 16-bit VL identifier.

Redundancy

An AFDX network is constructed so there are two independent paths (including MACs, PHYs, and cabling) between each ES, as well as redundant switches to protect the network from a failure at the MAC level or below (Figure 4). The default mode is to transmit the same frame (with identical frame sequence numbers) across both networks. The receiving ES then accepts the first valid frame and passes it to the application. Once a valid frame is received, any other frame with the same sequence number is discarded. The redundancy option must be configurable—frames for a given VL may be sent along either or both of the networks.

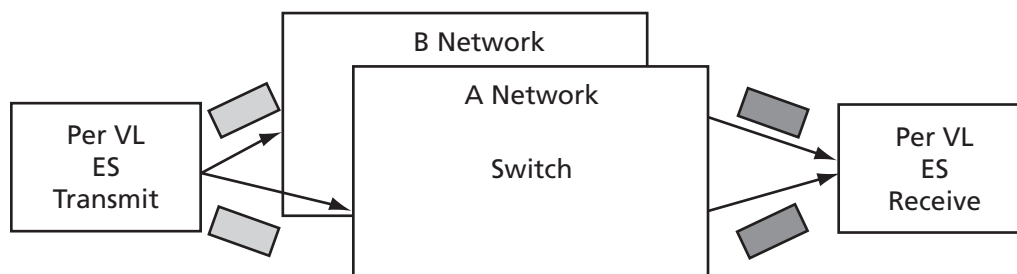


Figure 4 • Network Redundancy

Hardware Implementation

System Considerations

The one aspect of AFDX that should be kept in mind when architecting a solution is that AFDX is a *profiled* network. Every VL and its associated bandwidth is defined in advance and controlled through configuration files at both the ES and the switch. The designer does not have to develop a one-size-fits-all, plug-and-play solution. Local optimization is allowed based upon the needs of the application.

As there is no universal or standard configuration, each end equipment manufacturer will need to determine what the optimal solution is for the application at hand, taking into account what equipment from other vendors will need to interface with the Eses under design. The system architect has the choice of either defining a set configuration for all ESes or taking full advantage of programmable logic by defining a range of available configurations.

In addition, the system integrator will need to define the requirements for the switch, based upon the needs of each ES to be connected.

To ensure interoperability given the flexibility of AFDX/ARINC 664, vendors will need to coordinate with other vendors whose ESes will be communicating with theirs.

End System Design

Partitioning the Solution

Once the system architect has defined the required configuration, the designer must then decide which functions are best handled in hardware and which in software. The solution outlined here will take advantage of programmable logic devices and IP cores offered by Actel.

Figure 5 on page 7 illustrates the proposed solution. The major blocks contained in the FPGA are as follows:

- **CPU.** During transmission, the CPU is responsible for the ordering (scheduling) of frames to be sent. During reception, the CPU must determine the VL assignment for each incoming frame and distribute the frame to the appropriate VL queue. An ARM7 32-bit processor would be an appropriate host controller for this application.
- **MAC.** Two complete Actel Core10/100 Ethernet MACs are required to construct an ES, one for each channel. These two cores are compliant with IEEE 802.3 and ARINC 664.
- **ES Transmit.** This block contains the Regulator, BAG, and jitter timers needed to control the timing of frames sent to the MACs for transmission.
- **Transmit Redundancy Controller (TRC).** The TRC selects whether a frame will be transmitted on the Channel A PHY, Channel B PHY, or both.
- **Receive Redundancy Management (RRM).** This block is responsible for determining if a received frame is valid and for rejecting any duplicate valid frames received on the redundant channel.

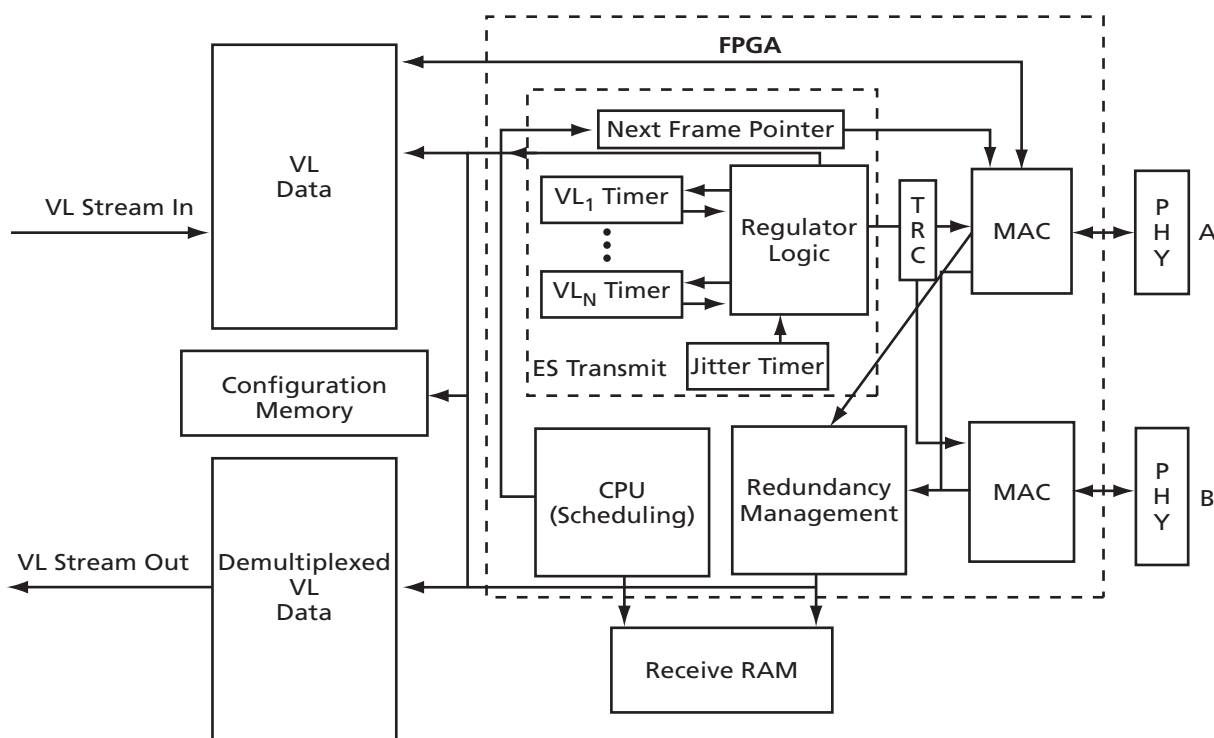


Figure 5 • End System Block Diagram

The following are the memory elements, which can be either internal or external to the FPGA:

- **Receive Data RAM.** This memory contains the RRM-processed received frames.
- **VL Received Data RAM.** This memory contains the demultiplexed (segregated by VL) received frames.
- **VL Transmit Data RAM.** This memory contains the frames to be transmitted via the network for each VL.
- **Configuration Memory.** This block contains all of the configuration parameters specific to the ES.

The size, configuration, and data format of each memory space is left to the system architect. The major functional blocks are discussed in further detail below.

CPU

During transmission, the host CPU is responsible for the ordering (scheduling) of frames to be sent. During reception, the CPU must determine the VL assignment for each incoming frame and distribute the frame to the appropriate VL queue. The host processor in an AFDX system will typically need to have a 32-bit data width to manage two message streams such as the Actel ARM7 processor core.

On transmission, the CPU must perform the following tasks:

- Configure the system at start-up
- Schedule transmit frames
- Demultiplexing of received frames

System Configuration

During start-up and reset, the CPU will read the configuration memory to determine the number of VLs, the bandwidth required for each, and the MAC addresses of the EEs. The CPU must then write to the appropriate register to control the Regulator and configure its own operation.

Scheduling

The CPU will inspect each incoming VL frame stored in the VL Transmit Data RAM to determine the optimal order for transmission (actual timing of frame transmission is left to the Regulator). This scheduling of the frames should be determined by a costing function that may be based upon one or more of the following parameters:

- The order of frames received for transmission
- The number of VLs to be serviced
- The length and VL assignment of the next *n* frames in the queue
- The loading of each VL queue
- The status of each VL's Free-to-Transmit flag (FTT)
- The value of each VL's BAG timer

The exact form of this costing function is left to the system architect to develop and will probably vary in complexity based on the number of VLs and the bandwidth requirements of the ES.

Once the next frame to be transmitted has been chosen, the CPU will update the Next Frame Pointer and alert the Regulator.

MACs

The MACs are responsible for transmitting data from the VL Transmit RAM, as defined by the Scheduler and controlled by the Regulator. On receipt, the MACs take data from the PHYs and deliver it to the Redundancy Manager.

To simplify the overall design task, this solution takes advantage of Actel-developed IP—Core10/100 Ethernet Media Access Controller (Figure 6). This IP block implements Carrier Sense Multiple Access with Collision Detection (CSMA/CD) algorithms as defined by IEEE 802.3 for media access control over Ethernet. Core10/100 has been tested both in simulation and in hardware and is currently flying in an AFDX application.

The core communicates with the host CPU via a set of Control and Status Registers and via the DMA controller for external shared RAM. Core10/100 operates as a DMA master for data transfers. It automatically fetches data from the transmit buffers and stores received data in external RAM with minimal CPU intervention. The linked-list management allows the use of various memory allocation schemes. External RAMs are used as configurable FIFOs with separate memory spaces for transmit and receive.

Communication with the off-chip PHYs occurs through the Media Independent Interface (MII).

The core has a generic host-side interface for connecting with the host CPU. This host interface can be configured to use an 8-, 16-, or 32-bit data bus width with big or little endian data.

For more information on Core10/100 functionality and use, refer to the *Core10/100 Ethernet Media Access Controller* datasheet. This document and an evaluation are available at <http://www.actel.com>.

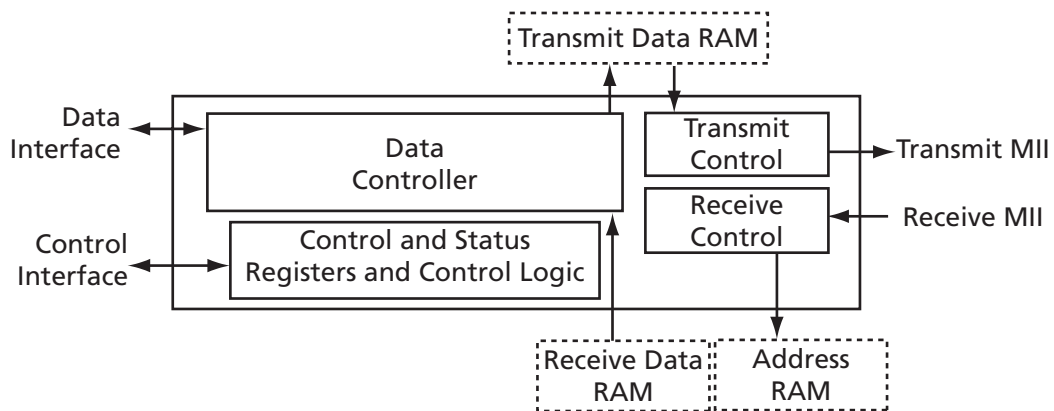


Figure 6 • Core10/100 Block Diagram

ES Transmit Block

The ES Transmit block contains the Regulator, BAG, and jitter timers needed to control the timing of frames sent to the MACs for transmission. The most complex part of this block is the Regulator. The Regulator must perform the following tasks:

- Send frames to the MAC for transmission, compliant with the bandwidth constraints set for a given VL and in accordance with ARINC 664
- Control the BAG and jitter timers
- Monitor both the BAG and jitter timers to set the FTT flags
- Alert the TRC as to the redundancy necessary for the next frame to be transmitted

Below are three proposed solutions for this sub-block. The optimal solution will depend upon the amount and type of traffic the ES will be required to handle.

Solution 1

The first proposed solution is to synchronize all VLs so that transmission occurs on 1 ms boundaries (the minimum BAG value), thereby simplifying the control logic and timing.

Each VL is assigned its own independent BAG timer. Even if two VLs have the same BAG value, they cannot use the same timer, as data for differing VL may be presented to the MAC at different times. BAG timers are free running and are held in reset only when the jitter timer times out.

This solution also employs a single jitter timer to indicate whether the frame can still be transmitted within the BAG for a maximum bandwidth signal. Once the jitter timer times out, it will be held in reset until the next 1 ms time slot.

The local control logic for each VL stream sets an FTT flag to indicate to the Regulator that a frame for this VL may be transmitted. [Figure 7 on page 10](#) shows the relative timing of these signals for two VL streams. VL_A has a BAG value of 2 ms, and VL_B has a BAG value of 4 ms. The following is a description of events, in order of occurrence:

- The outgoing transmission sequence starts with VL_{A0} (VL_A, frame 0), transmitted at 0 ms.
- At 1 ms, VL_{B0} is transmitted.
- At 2 ms, the FTT_A flag is set. After 2 ms, and after the BAG_A interval but within the jitter interval, VL_{A1} is sent and the FTT_A flag is reset.
- At 2 ms, the FTT_A flag is set.
- After 4 ms, and in the next valid BAG_A and jitter interval, frame VL_{A2} is transmitted. The FTT_A flag is reset.
- At 4 ms, the FTT_A flag is set.
- At 5 ms, VL_{A2} is still being transmitted but completes within the jitter interval. The FTT_B flag is set.
- After 5 ms and within the jitter interval, as VL_{A2} has completed transmission, VL_{B1} is transmitted. The FTT_B flag is reset.
- At 6 ms, the FTT_A flag is set.
- At 6 ms plus the jitter interval, as no frame for VL_A is ready for transmission, the BAG_A timer is held in reset.
- At 7 ms, a frame for VL_A is available, and VL_{A3} is transmitted. The BAG_A timer is released from reset, and the FTT_A flag is reset.
- At 9 ms, VL_{B2} is transmitted. The FTT_A flag is set.
- After 9 ms and within the jitter interval, VL_{A4} is transmitted. The FTT_A flag is reset.

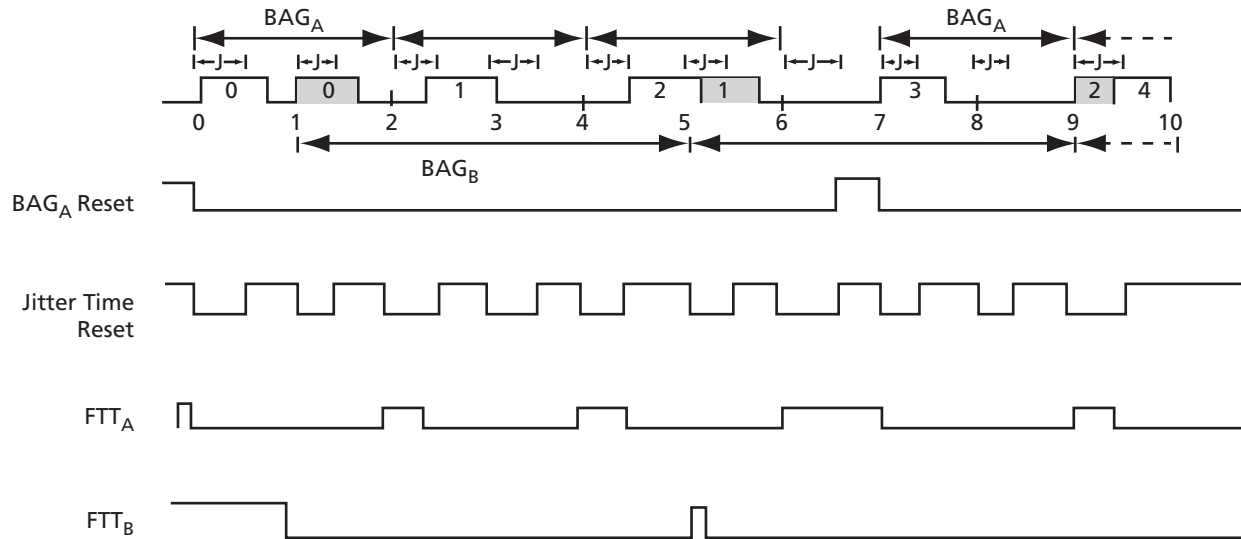


Figure 7 • ES Transit Timing

Note: In the actual design, FTT_A and FTT_B need to be set at each BAG interval. These are left out of Figure 7 for clarity. Since the allowed jitter interval is relatively long (more than 150 μs), a small part of the interval may be used for setting and handling flags.

On a polling cycle of 1 ms, the Regulator inspects the next frame in the transmit queue to determine to which VL the frame belongs (the last two bytes of the IP destination address). Once the VL is known, the Regulator checks the VL's FTT flag status. If the flag is set, indicating that the frame may be transmitted legally, the Regulator ascertains whether the MAC is ready to transmit the frame, and if so commands the MAC to transmit the next frame in memory. Upon start-up and reset, each FTT flag is set and each BAG timer held in reset until transmission of a frame is initialized via the associated VL.

This solution may consume less power than others, but it might be more appropriate for an ES with a small number of VLs or VLs with low bandwidth requirements.

Solution 2

An alternate solution to reduce system latency and increase throughput for networks operating at 100 Mbps involves creating four time slots for transmission. A complete AFDX frame with preamble and interframe gap will contain at most 1,538 bytes (minimum length is 84 bytes). A VL operating at maximum bandwidth (100 Mbps) would take at least 153 μs to transmit a frame of maximum length. Since the smallest BAG is 1 ms, creating time slots every 250 μs should allow for four channels.

A problem could arise when transmitting frames of maximum length—the transmission time plus the allowed jitter could exceed the allowed 250 μs time slot. To make each channel truly nonblocking, the designer would need to either shorten the allowed jitter interval or restrict the allowed frame size for each VL. Both approaches are allowed by the specification.

The system integrator would have to assign each VL to a time slot, the allocation being defined in the configuration file for the ES. This approach would allow an ES to handle four maximum bandwidth VLs, each at the maximum bandwidth, or allow for the segregation of higher bandwidth links from slower links.

This approach does add some complexity to the design:

- Four jitter timers would be required—one for each time slot/channel.
- For each frame, the Regulator would have to look up the time slot assignment for the associated VL.

The Regulator would need to operate at a higher polling rate (250 μs). Otherwise, the solution would function as described above.

An alternative would be to allow each frame to be transmitted in the next available time slot. This would not offer nonblocking channels but would do away with the time slot assignment, thus simplifying the Regulator logic.

Besides offering better throughput and system latency, this approach presents a more balanced load to the switch.

Solution 3

A third approach to the ES Transmit block is to allow the next frame in the queue to be transmitted as soon as the MAC is ready to transmit (assuming that the BAG restriction has been met). This requires that each VL have an associated jitter timer, thus increasing the amount of required logic.

At the end of each transmitted frame or during reset, the Regulator inspects the next frame in the queue to determine its assigned VL and then check that VL's FTT flag. If the flag is set, the Regulator verifies that the MAC is clear to transmit and then transmits that frame. This is similar to the alternate flavor of Solution 2 without the concept of time slots.

Though it adds complexity, such an approach would allow for the construction of a parametrized HDL core (for example, a core supporting n VLs with n BAG and jitter timers).

Transmit Redundancy Controller (TRC)

According to ARINC 664, transmit redundancy must be configurable on a VL-by-VL basis. As a result, the TRC will pass the frame to MAC A, MAC B, or both, based upon input from the Regulator.

Receive Redundancy Management

This block is responsible for determining if a received frame is valid and for rejecting any duplicate valid frames received on the redundant channel. This block, as shown in Figure 8, has two main tasks:

- Integrity Checking. For each receive channel, this block determines if the frame received has the expected sequence number for its VL.
- Redundancy Checking. This block passes only valid frames and deletes any duplicate frames.

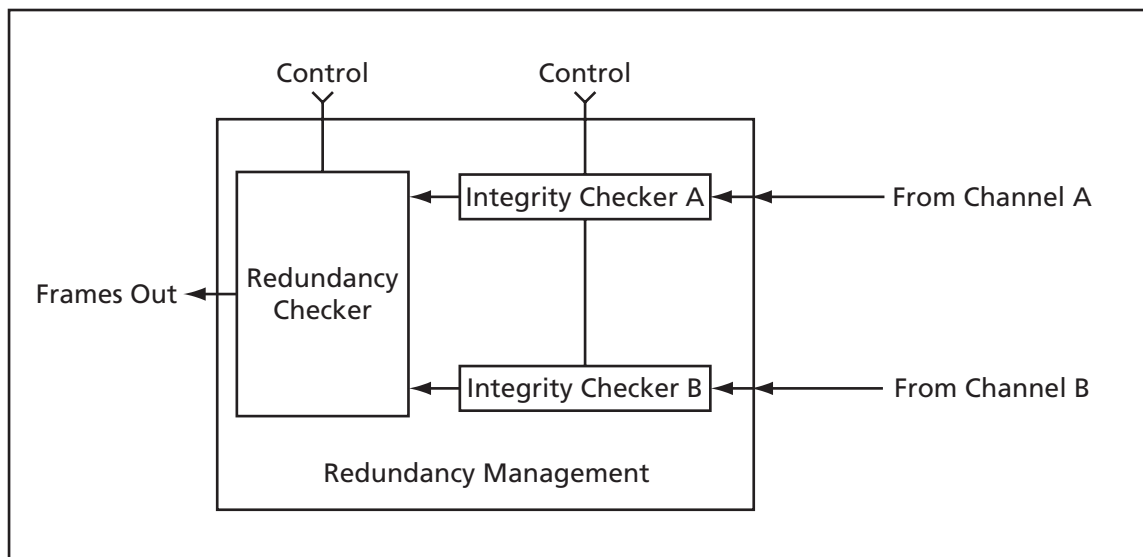


Figure 8 • Redundancy Management Block Diagram

Integrity Checker

For each received frame passed from the MAC, the Integrity Checker must verify that the frame received has the expected sequence number for its VL. The expected sequence number is the current frame number plus either one or two (taking into account that sequence numbers will wrap from 255 to 1). This check is based upon the last frame received, even if it was discarded.

Sequence checking allows for a single dropped frame in the link and is done on both channels in parallel, irrespective of redundancy settings.

There are special cases for valid sequence numbers that are out of order:

- A sequence number of zero will always be accepted, as it indicates a transmitting ES reset.
- Any frame sequence number will be accepted for the first valid frame received after an ES reset.

If the Integrity Checker encounters an invalid frame, it drops the frame and notifies the system of the error (the specification does not specify the form or format of error notification).

The specification requires that it be possible to disable integrity checking on a VL-by-VL basis. Integrity checking status is set at ES start-up via the configuration file.

Redundancy Checker

This function, when enabled, passes the first valid frame with a given sequence number for each VL. Valid frames are received via the Integrity Checker from Channel A, Channel B, or both. If the Redundancy Checker receives two copies of the same frame, it will pass the first valid frame and drop the second. No comparison is performed on the contents of the frame. When disabled, this block forwards all frames received on either channel. Redundancy checking can be disabled on a VL-by-VL basis.

The Redundancy Checker relies on an additional parameter, *SkewMax*, when accepting frames. *SkewMax* is defined as the maximum allowable time between reception of valid frames and is defined by the system integrator for each VL. If *SkewMax* for a VL is exceeded, the Redundancy Checker will reset both Integrity Checkers so they will accept the next valid frame, regardless of sequence number. This mechanism allows the ES to tolerate temporary data dropouts.

Switch Design

At the heart of an AFDX network is the switch, which has more responsibilities than its counterpart in a commercial Ethernet network. In addition to the obvious switching functions, an AFDX switch must perform frame filtering and traffic policing duties, ensuring that traffic arriving at the switch is compliant with the restrictions set for the appropriate VL. These additional requirements add complexity to the switch design.

In an AFDX network, each redundant network is connected through an independent switch. Each switch passes traffic without regard to redundancy, i.e., the two switches do not pass information to each other about received frame sequence numbers. This lack of redundancy checking simplifies the switch design.

Partitioning the Solution

Once the system architect has defined the required system configuration, the designer must then decide which functions are best handled at the ports and which by the central switching fabric. The proposed solution leverages the design work done for the ES, thereby simplifying the switch fabric design.

Figure 9 on page 13 shows the major blocks for the proposed switch solution. The various blocks are as follows:

- **Switch Fabric.** This is responsible for delivering frames to the appropriate ports, configuring the Switch Ports, and providing monitoring functions.
- **Switch Port.** The Switch Port is similar in functionality to an Line Replaceable Unit (LRU) ES without redundancy, but adds the frame filtering and traffic policing functions.
- **Switch ES. Maintenance Port.** A single-ended ES (i.e., with a single MAC and PHY) for use in configuring and managing the switch.
- **VL Received Data RAM.** This memory contains frames received at each port for processing by the switch fabric.
- **VL Transmit Data RAM.** This memory contains the frames from the switch fabric to be transmitted from each port via the network.

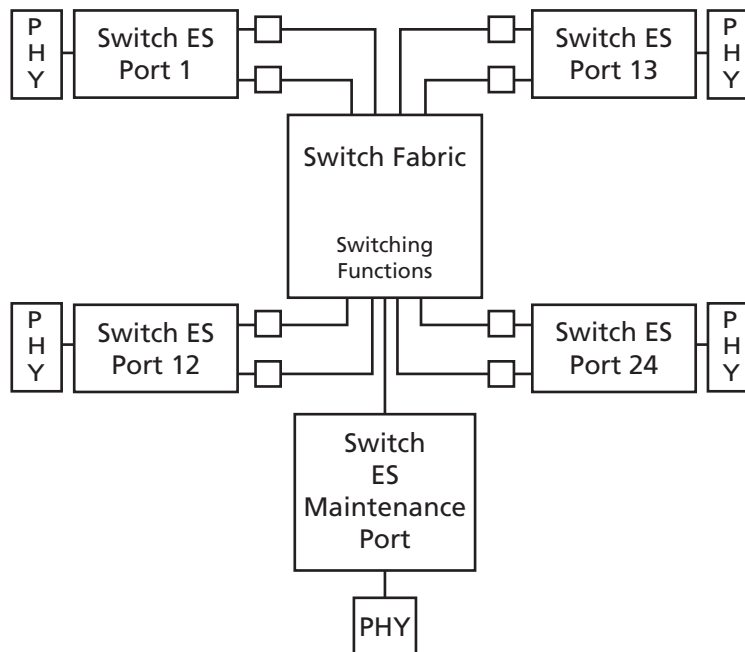


Figure 9 • Switch Block Diagram

Switch Port Design

Though ARINC 664 Part 7 does not contain the concept of a Switch ES, it does define the requirements of the Switch Port. Logic dictates that a Switch Port should resemble an ES in many respects.

The functions of a Switch Port are similar to those of an ES, with the exception of traffic policing and redundancy. A switch is required to police all traffic to ensure that each VL does not exceed its allotted bandwidth or violate its minimum and maximum frame sizes. A Switch Port must also perform integrity checking, but does not have to segregate incoming frames by VL, as the frames are switched in order of reception (allowing for Class of Service – refer to "[Switch ES Maintenance Port](#)" section on page 15).

On the transmit side, the port is required to transmit frames at line speed, respecting the InterFrame Gap. This requirement eliminates the scheduling and regulating requirements of the port.

A logical system partitioning, as shown in [Figure 10 on page 14](#), would place the policing functions inside the Switch Port, to be performed along with integrity checking. In this way, only policed data flows reach the Switch Fabric, allowing the complexity of the Switch Fabric design to be reduced. Since the proposed ES solution already includes a CPU, traffic policing and frame size checking may be added to the CPU's tasks, removing the VL segregation, redundancy management, and traffic shaping tasks.

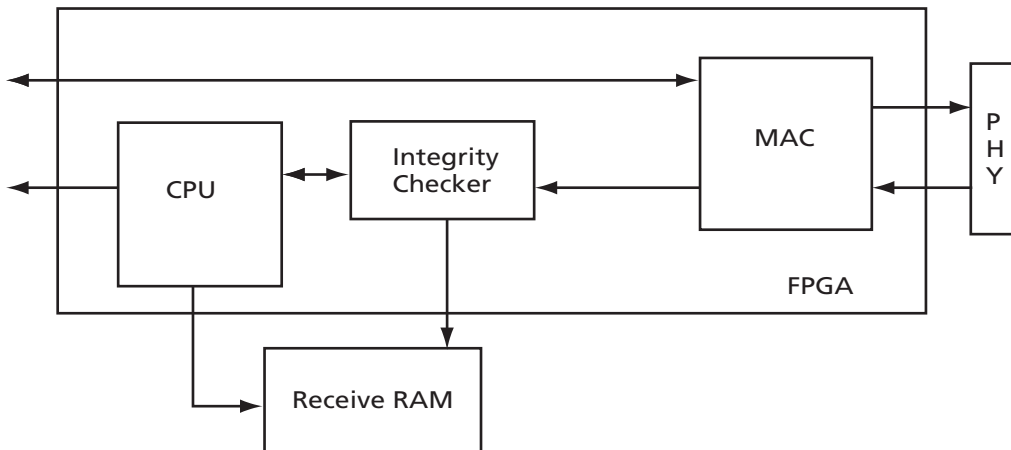


Figure 10 • Switch Port Block Diagram

Traffic Policing

ARINC 664 specifies a traffic policing function at the switch based upon the token-bucket algorithm common to switched-packet networks. The goal of traffic policing is to ensure that no VL exceeds its allotted bandwidth. From a timing perspective, a token-bucket algorithm may not be as strict as the outgoing traffic shaping function of the ES, allowing for "slop" in transmission over the physical layer.

The specification requires the use of one or both of the following two algorithms for traffic policing:

- Byte-based filtering
- Frame-based filtering

Since the proposed solution handles traffic policing post-MAC, frame-based filtering makes more sense.

Frame-based filtering is based upon comparing the balance of a token account to a reference when accepting or rejecting a frame. The token account is credited with tokens as time progresses, based upon the allowed average bandwidth for the link. When a frame is received, the account balance is compared to the reference value. If the balance exceeds the reference value, the frame is accepted and the account is debited by a calculated value. If the account balance is less than the reference value, the frame is dropped, an error is logged, and the account is not debited. These accounts are maintained independently for each VL.

For a given VL, an account AC is maintained and credited with tokens (in bytes) at the following rate (EQ 2):

$$AC = S_{MAX} / BAG \tag{EQ 2}$$

where:

S_{MAX} = L_{MAX} + Preamble + Start Frame Delimiter + Interframe Gap

L_{MAX} = the maximum allowed frame size for the VL

L_{MIN} = the minimum allowed frame size for the VL

BAG = Bandwidth Allocation Gap for the VL

Tokens accumulate in the account (AC) until it reaches the limit of

$$AC_{MAX} = S_{MAX} * (1 + [jitter / BAG]) \tag{EQ 3}$$

Where jitter is the defined jitter value for the VL

The specification does not describe how to determine or set the receive jitter value for a given VL but does restrict its range from 0 to 1 ms.

The switch configuration file contains the values for L_{MIN} , L_{MAX} , BAG, AC_{MAX} , Jitter, and Class of Service for each VL.

Once the CPU receives a frame from the Redundancy Management block, it does the following:

- Checks the size of the frame to ensure it is between L_{MIN} and L_{MAX} .
- Calculates a CRC for the frame, compares it to the Frame Check Sequence value contained in the frame, and discards the frame if there is a mismatch. The CRC calculation may be best handled in a logic block inside the FPGA.
- Inspects the value of AC and compares it to S_{MAX} . If the number of tokens is greater than S_{MAX} , the CPU passes the frame, and discards the frame otherwise.
- Writes accepted frames to the appropriate port's VL Received Data RAM.

If any error is detected, the frame is dropped and the error is reported to the Switch Fabric for inclusion in the Management Information Base (MIB). If there is a link failure, any frames that are held in the VL Transmit Data RAM should be discarded.

Switch ES Maintenance Port

The Switch ES Maintenance Port is used for communication with the switch for configuration and management functions. This Switch ES Maintenance Port is identical in functionality to an LRU ES in all respects except that redundancy is not implemented (i.e, it has one channel only).

Switch Fabric Design

Once frames have been validated and policed, they are written to the VL Received Data RAM for the appropriate port. The Switch Fabric is responsible for retrieving the frames and writing each to the correct output port while maintaining the ordinal integrity of the frames in each VL. If an output port is unable to accept a frame, the Switch Fabric should drop the frame to avoid hanging up switch traffic.

In addition, there is an aging function in the switch. Any packet older than a defined maximum delay should be discarded. Maximum delay parameters are defined for each port and are contained in the switch configuration file. The delay is defined as the interval between two events:

- Arrival of the last bit of a frame at the input port of the switch
- Delivery of the last bit of the frame from the destination output port

The specification does not define precisely where these events should be measured – PHY, MAC, or buffer. ARINC 664 Part 7 defines a traffic prioritization mechanism based on MAC destination address. There are two classes of traffic: high and low priority. Class of Service is set for each VL and is defined in the switch configuration file. The switch is responsible for writing high priority frames for a port before lower priority ones. However, the prioritization should not interrupt any low priority frame currently being transmitted.

The Switch Fabric is also responsible for monitoring functions as well, tracking network statistics and logging errors.

Architecting a detailed solution for the Switch Fabric is beyond the scope of this document.

Prototyping Support

To assist customers in prototyping solutions based on Actel FPGAs and IP, Actel produces a development board that can be used to evaluate an AFDX ES or Switch ES: the Platform8051 Development Board.

The Platform8051 Development Kit is intended as a demonstration and evaluation kit for a variety of Actel intellectual property, including Core8051, Core10/100, CoreSPI, CoreI2C, CoreSDLC, Core16X50, CoreUART, and the Utopia family cores (slight modification required). With its daughter card it is a versatile board that can demonstrate a variety of cores and customer designs.

The Platform8051 board contains the following:

- ProASIC^{PLUS} APA600-FG676
- 4 Mb SRAM Memory
- 32 Mb Flash Memory
- 10/100 Ethernet PHY (2 available)
- RJ-45 Port (2 available)

- RS-232 Serial Interface
- LCD display
- 10-bit ADC

All subsystems can be addressed by the APA600. In addition, there are several unpopulated component footprints on the Platform8051 board, including a location for a second Ethernet PHY that might be used to build an AFDX interface that could then be connected to AFDX test equipment to evaluate various design solutions.

Actel Device Advantages

As commercial aircraft have begun flying at higher altitudes, avionics designers have become aware of the occurrence of Single Event Upsets (SEU) in semiconductor devices and the related Soft Error Rate (SER) in SRAM devices. An SEU occurs when a heavy ion strikes a flip-flop or an SRAM cell. The incoming heavy ion can deposit enough charge to cause the flip-flop or memory cell to change state, corrupting the stored data. Because this occurs without permanently damaging the storage element, SEUs are often referred to as soft errors.

SRAM FPGAs present a special class of devices when discussing the impact of SEUs on reliability and functionality. In contrast to antifuse and Flash-based FPGAs, SRAM FPGAs also store their logic configuration in SRAM switches. These configuration switches represent more than 90% of the total SRAM bits in an SRAM FPGA. So in addition to the susceptibility of their sequential logic cells and embedded SRAMs to SEUs, SRAM FPGAs are also susceptible to configuration upsets wherein the routing and functionality of the circuit is corrupted (even I/O configuration can be affected). These types of errors are extremely difficult to detect and correct and are practically impossible to prevent.

Configuration upsets can result in system failures, in some cases with catastrophic consequences if the upset creates a short damaging the device or board, or causes the system to operate in an erratic and unpredictable way. This susceptibility of SRAM FPGAs to configuration errors has resulted in new terminology – the single event functional interrupt or SEFI.

Given that the focus of AFDX is on reliable communication, it is only natural to select reliable components during the design process. All Actel FPGAs are immune to this class of upset.

Moreover, all Actel FPGAs are single-chip solutions (i.e., no configuration PROM or other external support chips are required). This characteristic reduces board space and weight, critical design parameters in flight hardware. In addition, all Actel FPGAs are live at power-up, eliminating the lengthy "reboot" time required for an SRAM FPGA during a system reset that could occur while in flight.

The combination of these three advantages—SEFI immunity, single-chip solution, and live at power-up—makes Actel FPGAs well suited as a base for AFDX interface design. In addition to providing the industry's best Aerospace FPGAs, Actel has extensive experience supporting Aerospace customers through a world-wide sales team. For more information on Actel support services or to locate the sales office nearest you, visit <http://www.actel.com>.

Summary

AFDX (ARINC 664, Part 7) represents a major upgrade in both bandwidth and capability for aircraft data networks used in commercial aircraft. Its reliance on COTS Ethernet technology helps to lower some of the implementation costs, though the requirement for guaranteed service does present challenges to system designers.

This document has covered potential solutions for constructing AFDX ESes and Switch ESes using Actel FPGAs combined with Actel-provided IP cores. In addition, requirements have been outlined for constructing a Switch Fabric, which is central to an AFDX network.

The advantages of Actel FPGAs for avionics applications, combined with Actel-provided IP and development platforms, give engineers an excellent solution for designing AFDX interfaces that can not only be used in commercial avionics, but may well have other applications in industry, where reliability, speed and guaranteed service are required.

References

1. ARINC Specification 664, Part 1, Aircraft Data Network, Systems Concepts and Overview. 31 January 2002.
2. ARINC Draft 3 of Project Paper 664, Aircraft Data Network, Part 7, Avionics Full Duplex Switched Ethernet (AFDX) Network. 13 September 2004.
3. CES White Paper on AFDX, Creative Electronic Systems S.A. 4 November 2003.
4. Quality of Service Solutions Configuration Guide, Cisco Systems, Inc. 2000
5. Frame Relay Traffic Shaping, Cisco Systems, Inc. 2003

Related Documents

Datasheets

Core10/100 Ethernet Media Access Controller

http://www.actel.com/ipdocs/Core10100_DS.pdf

Actel and the Actel logo are registered trademarks of Actel Corporation.
All other trademarks are the property of their owners.



www.actel.com

Actel Corporation

2061 Stierlin Court
Mountain View, CA
94043-4655 USA

Phone 650.318.4200
Fax 650.318.4600

Actel Europe Ltd.

Dunlop House, Riverside Way
Camberley, Surrey GU15 3YL
United Kingdom

Phone +44 (0) 1276 401 450
Fax +44 (0) 1276 401 490

Actel Japan

www.jp.actel.com

EXOS Ebisu Bldg. 4F
1-24-14 Ebisu Shibuya-ku
Tokyo 150 Japan

Phone +81.03.3445.7671
Fax +81.03.3445.7668

Actel Hong Kong

www.actel.com.cn

Suite 2114, Two Pacific Place
88 Queensway, Admiralty
Hong Kong

Phone +852 2185 6460
Fax +852 2185 6488