ISSN 2706-655X Vol.5, Issue 7: 1-6, DOI: 10.25236/IJFET.2023.050701

Design of ARINC664 switch based on FPGA

Xu Siqi[1,a]

[1]*China Flight Test Research Institute, Xi'an, Shaanxi, China*
[a]*873079269@qq.com*

Abstract: *ARINC664 switch based on FPGA was designed to meet the requirements of engine test communication in aero-engine flight test bench, which was used for the network data interaction between flight data transfer platform, engine controller and health management system. The design principle of ARINC664 switch is introduced. The hardware logic of each module inside the switch is designed in detail, including ARINC664 switch module based on FPGA, ARINC664 in-band terminal module, etc. At the same time, the design flow of embedded software of switch is given. It can effectively ensure the real-time, operational, network reliability and security of ARINC664 bus data interaction between the flight platform data transfer platform and the tested engine, and has certain engineering practical value.*

Keywords: *Flight platform; Throttle control system; Throttle controller*

1. Introduction

The ARINC 664 network standard, developed by the American Aeronautical Radio Corporation, is a communication protocol standard for full-duplex switched Ethernet dedicated to aviation [1,2]. It has the characteristics of high bandwidth bottom delay, determinism, double redundancy and high reliability. ARINC 664 is the communication mode between the data transfer platform of an aircraft engine flight platform and EEC and EMU of the tested engine. The number of ARINC 664 required by EEC and EMU is 6, and dedicated links for EEC, EMU and online parameter data should be set to prevent interference between data links. Therefore, the ARINC 664 bus data switch was added to the flight station[3,4]. The overall data interaction relationship between flight station ARINC 664 bus data switch and flight station is shown in Figure 1 below. ARINC 664 bus data switch is used to connect EEC, EMU and flight station data transfer platform[5].

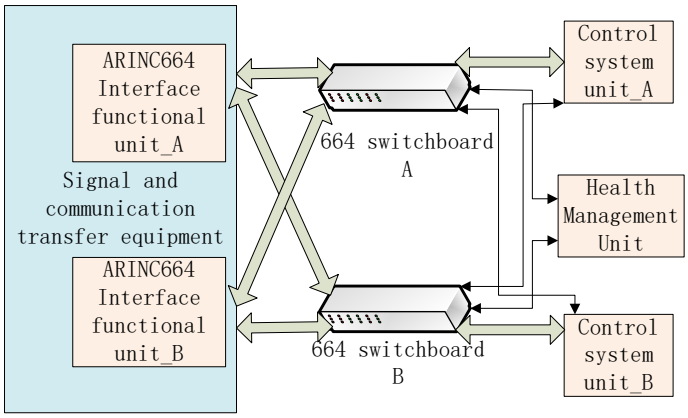

Figure 1: Interaction diagram

2. Working principle of the switch

The ARINC664 switch mainly implements data exchange and data communication for 8 channels of ARINC664 data. The CPU processor module adopts FT2000, and the operating system adopts domestic operating platform, which is responsible for the management and configuration of the switch network, the configuration of the in-band ARINC664 port and the management of data sending and receiving. The internal data exchange core is realized by a XilinxK7 FPGA, on which 8 MAC controllers and 1 built-in terminal system function interface are embedded. The ARINC664 data cache

ISSN 2706-655X Vol.5, Issue 7: 1-6, DOI: 10.25236/IJFET.2023.050701

is implemented by an external DDR. The FPGA accesses the ARINC664 network through the embedded MAC controller, processes, queries, analyzes and forwards the received data, and finally forwards it through the MAC controller on the FPGA. The in-band ARINC664 terminal uses a XilinxK7 FPGA to connect to the functional port of the internal end system of the switch chip. The ARINC664 switch uses J599 connectors to connect to the ARINC664 network.

2.1. CPU

As the core processor, the CPU module is the core module for switch management. It is connected to the host configuration management software through network ports, and to the switch FPGA module and terminal FPGA module through PCIE ports. In addition, the CPU module provides a debugging serial port.

2.2. Clock Module

The whole board provides one 100MHz clock signal, and the logic module provides it to other logic modules after clock frequency division. mii_clk provides the PHY clock, clk_boot is the remote load module clock, clk_sys is the master clock for each module in the system clock logic, and clk_ddr is the clock used by the DDR module.

2.3. Switch Module

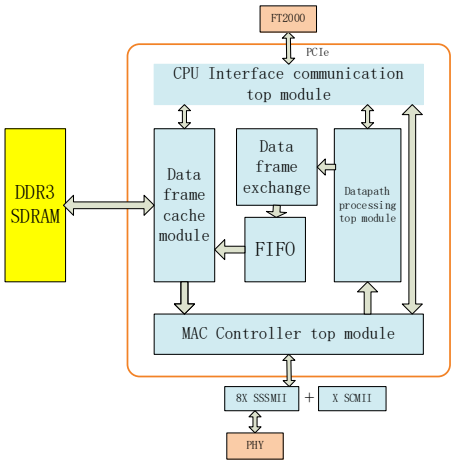

Figure 2: Overall functional block diagram of switch module FPGA

The overall functional block diagram of the switching module FPGA is shown in Figure 2.The main function of the CPU interface communication top-level module is to convert the PCIe bus timing of the CPU interface into the communication timing realized in the FPGA, and to execute various configuration and control information of the CPU through the internal transmission path in the FPGA. Any register space on the FPGA can be accessed via the PCIe bus on the CPU side.

The top layer module of the MAC controller connects to the ARINC664 data link layer in the interface FPGA. Nine MAC controllers are embedded inside the module, eight of which are connected to the PHY chip outside the FPGA via the SSSMII interface, and one is connected to the internal in-band ARINC664 terminal. All nine ports enable full-duplex 10/100Mbps data communication. The MAC controller of each corresponding port needs to collect statistics on the data frames received by the port, and provide MIB statistics counters for the upper-layer SNMP NMS software.

The main function of the top layer of data path control module is to insert the data frames received from the MAC controller into the ARINC664 switch internal identification label according to the requirements of the VL lookup table configured in DDR3 SDRAM.

The main function of the top layer module of the data frame exchange engine is to receive the data frame sent by the top layer module and control the data frame with the internal identification label of the AFDX switch. According to the internal identification label, the module forwards the data frame to the corresponding FPGA port.

The main function of the data frame cache top-level module is to receive the data frames sent by the

ISSN 2706-655X Vol.5, Issue 7: 1-6, DOI: 10.25236/IJFET.2023.050701

top-level module of the data exchange engine, and store the received data frames to the off-chip DDR3 SDRAM in the order of sending port and high priority.

2.4. Terminal Module

The terminal module completes the main protocol processing process of data receiving and sending, and the overall functional block diagram of FPGA logic is shown in Figure 3. These modules include PCIe interface modules, user port interface modules, IP fragment sending module, IP recombination receiving module, traffic sending control module, data cache control module, and redundancy management module.

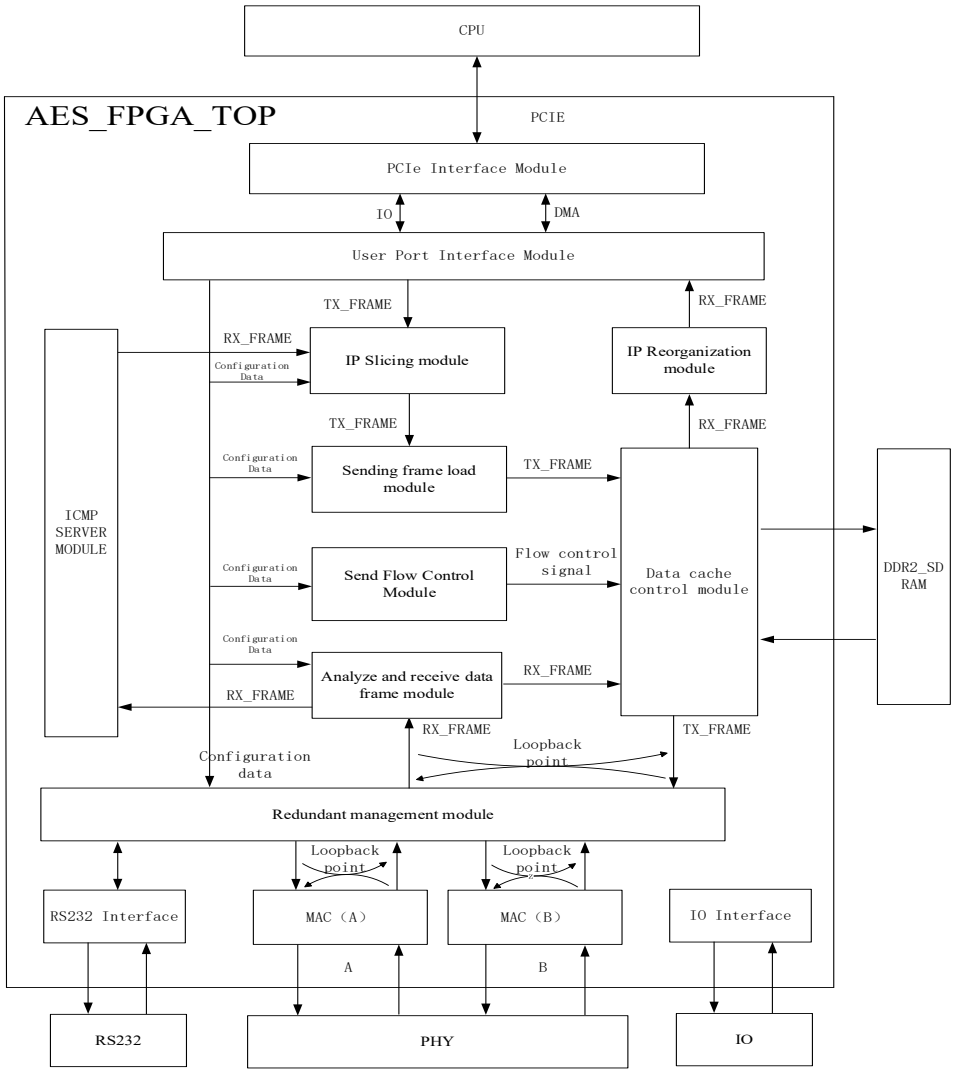

Figure 3: Overall functional block diagram of FPGA terminal module

A PCIe interface module connects a CPU to a terminal NIC for data communication. The host driver and PCIe interface module work together to complete the bidirectional DMA transfer and IO register read and write operations between the terminal device and the host.

The main function of the user port interface module is to connect the PCI interface module and other functional modules for register and DMA communication. The PCIe interface module receives the control commands sent by the host in register read/write mode, forwards the commands to the corresponding function module, and reports the NIC status and MIB information to the host through the PCIe interface module. The module also needs to complete bidirectional DMA data transfer between the PCIe interface module and the data cache control module.

The sending IP fragment module receives the load of data frames sent from the interface module of the user port and the ICMP reply data frames sent from the ICMP server module. It searches the sending configuration table according to the Port_ID of the sent data frames to find the maximum

ISSN 2706-655X Vol.5, Issue 7: 1-6, DOI: 10.25236/IJFET.2023.050701

length of data frames corresponding to the sending port. After finding the maximum data frame length, the module needs to IP fragment the current data frame according to the maximum frame length. In RFC791, the length of the IP fragment data frame, except for the last piece, needs to be a multiple of 8.

The function of the receiving IP reassembly module is to receive the data frame sent by the data cache control module and remove the invalid bytes in the data frame according to the IP fragment information in the internal identification label of the data frame. (Since the data cache control module adopts a burst reading mode of 128BIT for DDR2, when reading the IP fragment data frame, the data cache control module can delete the invalid bytes in the data frame.) There are 1-3 invalid 32BIT data between two IP fragments), which provides the user port interface module with a complete data frame after IP reorganization. The content of IP fragment information refers to the definition of the internal identification label of the data frame in the interface timing of the receiving IP reassembly module.

The main function of the sending flow control module is to realize the traffic shaping control of the sending VL by polling the internal RAM time counter. At the same time, the module needs to check the SN of the VL where each data frame is sent according to the sequence in which data frames are sent.

The function of the received data frame module is to receive the data frame from the downstream redundancy management module, and find the Port ID corresponding to the data frame according to the quintuple through the internal RAM, and then look up the Port type according to the Port ID. The Port ID and port type are added to the data frame internal label and sent to the cache management module. Analyze the received data frame module needs to analyze the IP header information of the data frame to determine whether the data frame is a data frame that has been IP fragmented. If the data frame is IP fragmented, the module needs to analyze whether the IP fragment information is correct. If the IP fragment information is incorrect, the module needs to filter the current data frame. When analyzing the IP header information, the receiving data frame module needs to determine the protocol loaded by the IP data frame and process the data frame according to different protocol types.

The function of the data cache control module is to manage and schedule the use of off-chip DDR2 SDRAM cache, cache the data frames of the sending PORT PORT and the receiving port port, and provide the corresponding data frame input and output statistics for the host and the sending flow control module.

The redundancy module provides redundant control of the direction of two data frames: the sending port and the receiving port. In the normal working mode of the terminal network card, in the direction of sending data frames, the redundant module receives the data frames sent by the sending cache module, and sends the data frames to the external network card through two different MAC controllers according to the marks of the A and B networks within the data frames. At the same time, the sending redundancy module also needs to fill the INTERFACE_ID of the sending port in the corresponding position of the packet header of the data frame according to the 664P7 protocol. In the direction of receiving data frames, the redundancy module mainly completes three functions, the first is to check whether the current VL_ID is in the receiving configuration table, the second is to perform data frame integrity check according to 664P7 protocol, and the third is to perform AB network redundancy check according to 664 P7 protocol.

3. Software Design

The software design flow of ARINC664 switch is shown in Figure 4 below. The FPGA design is loaded first after the CPU board of the switch is powered on or reset. The loaded FPGA design file is saved in a compressed format in the Flash file system. The loader first decompresses the FPGA design file, and the extracted file is temporarily saved in the memory file system (/ramdisk). After the decompression is complete, the loader triggers the logic to enter the loading state, and the loader writes in a single-byte write cycle until all the FPGA design files are written. Wait for the configuration confirmation. There is a timeout mechanism for waiting for confirmation. For example, the loading fails due to timeout and exits.

ISSN 2706-655X Vol.5, Issue 7: 1-6, DOI: 10.25236/IJFET.2023.050701

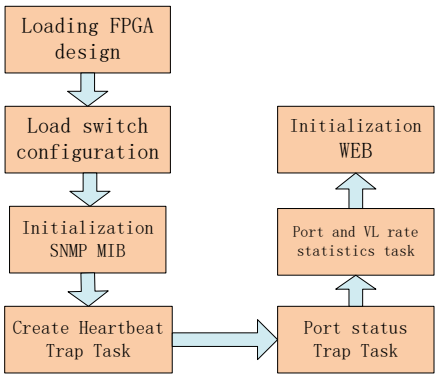

Figure 4: Start the process

After the switching logic is formed, the CPU uses the PCIe bus to configure the ARINC664 configuration table. After the configuration is loaded, the switch can forward data. Configuration upload is provided by the operating system Web Services HTTP POST file transfer interface, the user can register their own upload path through the httpUploadConfAdd function. The file upload function used in the WEB management of switches uses the same upload path. The HTTP POST request path is /upload, and the CPU board file system path is /ramdisk.

Configuration loading process during power-on reset and dynamic configuration update, intConnect is invoked by the operating system to register the interrupt handler. The port statistics task, VL statistics task, and port connection status task are stopped first. Because when the switch is running, the network management workstation may be in the port rate, VL rate statistics monitoring. To ensure data consistency, send an SNMP TRAP message to the network management workstation to stop monitoring statistics before configuration updates. Notify the monitoring operation again after the configuration update is complete.

The SNMP module initializes the MIB library and creates heartbeat trap tasks, port status trap tasks, and port and VL rate statistics tasks. The TRAP message is sent in the form of a task. The Trap message waits indefinitely on the message queue and transmits the port connection status message through the message queue. The port and VL statistics function includes port rate calculation tasks, VL rate calculation tasks, and MIB query statistics.

The WEB module creates a listening task for port 80 and enters the waiting state for user requests. Switches are managed through out-of-band Ethernet interfaces. You can set this parameter using the interface management on the WEB UI. Basic Settings include IP address and subnet mask. This project is saved in NVRAM.

The BIT self-check test includes power-on self-test, periodic self-test, and maintenance self-test. The self-check items include DDR status, PCIe status, and network port Link status. After the device is powered on, it checks peripheral components and reports the detection information to the CPU.

4. Conclusion

When a certain type of engine is tested on the aero-engine flight bench, the data transfer platform uses ARINC664 bus to communicate with the tested engine controller and other devices. The bus data switch is developed to network the installed platform with the controller and other devices for data interaction. On the basis of introducing the working principle of the switch, the hardware logic of each module of the switch is analyzed, including ARINC664 switch module based on FPGA, ARINC664 in-band terminal module, etc. The workflow of embedded software of switch is designed in detail. The real-time, operational, network reliability and security of ARINC664 bus data interaction between the flight platform data transfer platform and the tested engine are effectively guaranteed.

References

[1] Zhang Shuyan, Zhang Zhe, Yin Kai, et al. Research on Test technology of ARINC664 Network terminal System [J]. Automated Test Technology, 2013, 21 (9): 2419-2424.
[2] Li Chao. Design of aeronautical Communication System based on AFDX technology [D]. Harbin University of Science and Technology, 2009.

ISSN 2706-655X Vol.5, Issue 7: 1-6, DOI: 10.25236/IJFET.2023.050701

[3] Wang Weipeng, Zhang Yanyuan, Yao Jinhua. Design of ARINC664 /AFDX terminal system based on FPGA [J]. Microprocessors, 2009, 4(2): 1-4
[4] Shen Lei, Pan Chaojun. Design and Improvement of ASIC Chip for AFDX Switch [J]. Civil Aircraft Design and Research, 2013, 1674-9804.
[5] Zhao Yongku, Li Zhen, Tang Laisheng. AFDX Network Protocol Research [J]. Computer Measurement and Control, 2012, 20 (1): 8-11.