

Implementation of AFDX Switch on ZYNQ FPGA

T. Suresh¹, C. Ramesh Reddy², Y. Syamala³, A.V.N. Tilak⁴

Department of Electronics and Communication Engineering, GEC, Gudlavalleru, Andhra Pradesh, India

Abstract: Avionics Full-Duplex Switched Ethernet Network (AFDX), originally The Word “Avionics” is the combination of the Aviation and electronics, which could be defined as electronics of aircrafts, artificial satellites and spacecrafts. MIL-1553 as its communication protocol for Avionics projects. But MIL-1553 communication is half duplex, asynchronous and works at 1Mbps which may not be able to cater to the future communication performance requirements of bandwidth and maintainability. So a need exists to improve bandwidth, reliability and maintainability while at the same time reducing the physical dimensions of size, weight and number of connectors. A new network infrastructure is needed to provide more flexibility in avionics system design and cope with the increasing number of interconnections between systems. The proposed project is implementation of 8-port AFDX switch on ZYNQ FPGA and also to form a deterministic network means that all the AFDX End systems connected through AFDX switches via Virtual links, Finally the utilizes data speed is 100Mbps full-duplex switched Ethernet according to ARINC-664Part7. Merits of this project is redundancy can be achieved i.e. there is no repetition of frames at output of end-system and another one is complexity of the network can be reduced. This entire work is implemented in Xilinx Vivado 2014.4 and Xilinx 14.6 ISE simulator and the packet flow at the end-system can be captured by Wire-shark software.

Keyword: ARINC 664P7, MIL-1553, ZYNQ FPGA, AFDX

1. Introduction

The Word “Avionics” is the combination of the Aviation and electronics, which could be defined as electronics of aircrafts, artificial satellites and spacecrafts. The Scientist Mr.Collinson mentions that the term avionics “was first used in the United State of America (USA) in the early 1950s and has since gained wide scale usage and acceptance” simply it is Avionics Full-Duplex Ethernet network protocol. The paper is organized as follows. AFDX protocol is given in section II. Section III provides AFDX overview, while Section IV deals with previous techniques. Section V deals with proposed project and finally Section VI deals with simulation results and end test setup results and Section VII concludes the paper and provides guidelines for future scope.

2. AFDX Protocol

Aircraft Data Networks (ADN) primarily utilizes the ARINC 429 standard. This standard, developed over thirty years ago and still widely used today, has proven to be highly reliable in safety critical applications. ARINC 429 networks, which can be found on a variety of aircraft from both Boeing and Airbus companies, utilize a unidirectional bus with a single transmitter and up to twenty receivers. A data word consists of 32 bits communicated over a twisted pair cable. There are two speeds of transmission: high speed operates at 100 Kbit/s and low speed operates at 12.5 Kbit/s.

Another standard, ARINC 629, introduced by Boeing for the 777 aircraft provides increased data speeds of up to 2 Mbit/s and allowing a maximum of 120 data terminals. ARINC 629 network operates without the use of a bus controller, thereby increasing the reliability of the network architecture. One of the primary draw-back of this network [1] type is that it is very specific to civil aircraft applications, requiring custom hardware which can add significant cost and development time to the aircraft.

ARINC664 Part7 [2] is defined as the next generation aircraft data network (AFDX). It is based on IEEE 802.3

Ethernet, enabling greater potential use of Commercial Off-The-Shelf (COTS) hardware, thereby reducing aircraft cost and development time. AFDX was developed by Airbus Industries for the A380, has since been accepted by Boeing and used on the Boeing 787 Dream liner, and is being used or considered today for other applications [3].

3. AFDX Overview

In this section deals with about overview of AFDX and switch description and end system specification, and virtual link concept and also the brief explanation of Jitter and bag concepts..etc. Mainly AFDX network consists of three parts

- 1) AFDX Switch
- 2) AFDX End system
- 3) AFDX Virtual links

An aircraft data network has been described elsewhere in this standard as a profiled version of an IEEE 802.3 Ethernet utilizing IP addressing and related transport protocols. ARINC664 Part7 describes a subset of this network, where quality of service including timely delivery is paramount. The AFDX network is a special case of a profiled network. A deterministic network [4] may communicate with a wider profiled network and by inference, with a compliant network through routers or gateways. Figure1 depicts this network hierarchy.

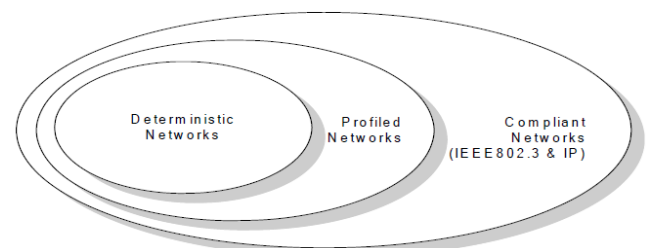


Figure 1: AFDX Network Hierarchy

3.1 AFDX Switch

The AFDX switch consists of five functional blocks that interact with each other, as shown in figure 2

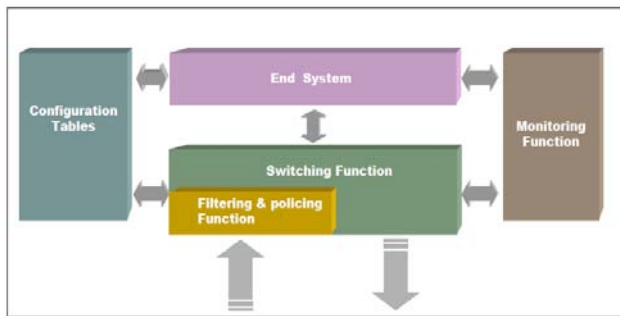


Figure 2: Functional elements of AFDX Switch

All frames arrive at the switch in the Filtering & Policing function stage where they are filtered in various steps. That apply rules about frame integrity, frame length, traffic budget and acceptable destination(s).

The core of the switching activity is performed by the switching function. Frames filtered by the filtering and policing function are forwarded to the appropriate physical output ports where they leave the switch again.

The above all functions are controlled by configuration data contained in static configuration tables. The end system stage provides the means to communicate with the switch (receives frames dedicated to the switch and allows the switch to send frames). This is used for data loading and monitoring functions.

All operations are monitored by monitoring functions that logs events such as the arrival of a frame or a failed CRC check and additionally creates statistics about the internal situation. Since the switch is a part of a network, it communicates with the network management function for operational information and for health related information. The frame size value shown in figure3

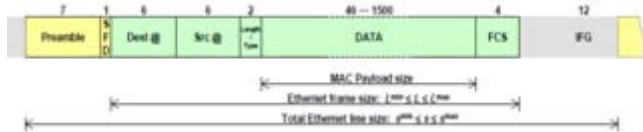


Figure 3: Frame size values

The actual time a frame occupies the Ethernet line (s). Therefore, all fields have to be taken into account: IFG (12 octets) + Preamble (7 octets) + SFD (1 octet) + MAC Frame size ($L = 64$ to 1518 octets).

3.1.1 Filtering function

The filtering causes the switch to distribute only valid frames to selected destinations. Upon arrival in a switch, each frame is examined and the contents of certain fields of the frame header (e.g. destination address field, frame check sequence field, etc.) and the construction of the frame itself are monitored:

- The frame size: whether the frame is either longer or shorter than the envelope allows.

- The frame integrity: whether the FCS embedded in the frame matches the calculation upon reception.
- The frame path: whether the destination requested by the content of the Destination Address field (which in case of the AFDX is the Virtual Link Identifier) of the arriving frame is permitted or not.

If the frame properties do not comply with the configuration parameters, the frame is filtered i.e. discarded, and one or more MIB entries updated. The definition of the MIB entries as well as the update conditions and processes are specified in the relevant specification document.

The following aspects of the frame are tested as part of the filtering function:

- Destination address validity (Ethernet Address corresponds to a valid VL, including constant field)
- This VL is valid to be received on that destination port (according to the switch configuration table)
- Frame Check Sequence validity
- Ethernet frame size (L) is an integral number of octets (alignment)
- Ethernet frame size (L) in the range [64 octets, 1518 octets]
- Ethernet frame size (L) less than or equal to L_{max}
- Ethernet frame size (S) greater than or equal to S_{min} , only in case where byte-based traffic policing is used

3.1.2 Traffic policing

This section describes a model of an algorithm that performs traffic based policing on the Destination Address basis. The Destination Address field contains the information that is used to identify a Virtual Link in an AFDX environment. A Virtual Link defines a traffic flow that has certain properties such as a group of recipients, or a minimum allowed gap between two frames. This traffic flow has to be maintained in a segregated way to guarantee its associated properties. In order to use in the context of this specification the same terminology as the one used in commercial documents the Destination Address is used synonymously with the term Virtual Link or "VL."

The jitter phenomenon is dependent upon Virtual Link and switch characteristics, as it is a function of the traffic of the total of all Virtual Links arriving at a particular switch. If a Virtual Link spans several switches, the actual jitter may be different on each of the intermediate switch. However, maximum values for latency and jitter for any switch provide an upper bound.

The traffic-policing model is described from an End-System point of view because End-Systems are the main traffic generators into the switch, shown in Figure 4. Model based description of the switch provides a switch centric view of the properties expected from a proper implementation of the switch. The switch may implement one of the two algorithm, either Byte-based or Frame based, or both of them. The choice of the algorithm will have an impact on the method used to prove the schedulability of the network.

The Switch should have a traffic prioritization mechanism based on MAC destination address with 2 traffic classes:

High Priority[5] and Low Priority. The priority level should be defined in the configuration table on a Virtual Link basis.

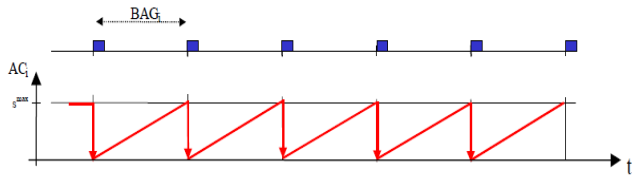


Figure4: Example of Traffic without Jitter

Description of the policing algorithm:

Initially, the ACcount for VLi (also called ACi, expressed in bytes) is set to

$$Simax (1+Ji/BAGi) \quad (1)$$

- ACi is checked every time a frame of VLi arrives in the switch.
- Let s be the total Ethernet line size of the received frame (S = Ethernet frame size (L) + 20 octets; the 20 octets correspond to IFG+Preamble+SFD).

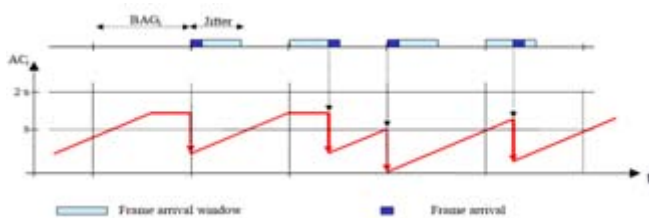


Figure 5: Example of Traffic with Jitter = BAG/2

A traffic policing mechanism should be implemented on the switch in order to ensure fault containment function of the network. Since a failed End System must not disturb the network, any frame belonging to a traffic flow that is not compliant with the network configuration should be discarded. The Configuration table has a relationship between MAC destination address, ACi, BAG, Jitter, Smax, and eventually Smin in the case where byte-based traffic policing is used. Traffic policing should be based on parameters: BAG, Jitter, Smax, and eventually Smin in the case where byte-based traffic policing is used. For each VL or group of VLs sharing the same account, the Traffic Policing function should authorize one BAG value according to the configuration table.

The Traffic Policing function of the switch should at least be configurable for BAG values in the range 1ms to 128 ms. For each VL or group of VLs sharing the same account, the Traffic Policing function should authorize one maximum Jitter value, according to the configuration table. The traffic policing function should at a minimum be configurable for maximum allowed Jitter values in the range 0 to 10 milliseconds. This implies that ACi has a maximum size of at least:

$$AC_{imax} = Simax (1+Ji/BAGi) \quad (2)$$

The traffic policing function should be able to handle at least Ethernet frames sizes (L) in the range [64-1518] octets.

3.2 End System

The main function of the End System (ES) is to provide services, which guarantee a secure and reliable data

exchange to the partition software. Quality of Service (QoS) provides a method for categorizing traffic and for ensuring that particular categories of traffic will always flow across the network at the service level to which they are entitled, regardless of competing demands.

3.3 Virtual Link

Description of the “Virtual Link [6]” concept is presented in Figure 7, since it is widely used in this project. An end system may be designed to only receive VLs and not transmit VLs, or the contrary; therefore, an ES can originate or receive zero VLs. End-systems exchange Ethernet frames through VL. Only one End System within the Avionics network should be the source of any one VL. A Virtual Link is a conceptual communication object, which has the following properties:

- A Virtual Link defines a logical unidirectional connection from one source end-system to one or more destination end-systems, shown in Figure 7.

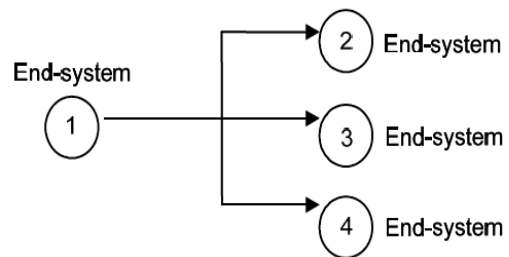


Figure6: virtual link path

- Each Virtual Link has a dedicated maximum bandwidth. This bandwidth is allocated by the System Integrator. The ES should provide logical isolation with respect to available bandwidth among the Virtual Link(s) it supports. Regardless of the attempted utilization of a VL by one partition, the available Bandwidth on any other VL is unaffected.

For each Virtual Link, the End System should maintain the ordering of data as delivered by a partition, for both transmission and reception (ordinal integrity).

3.4 Scheduling

In a transmitting end system with multiple VLs, the scheduler [7] multiplexes the Different flows coming from the regulators, as illustrated in figure 7, it shows how unregulated flow of packets converted in to uniform flow.

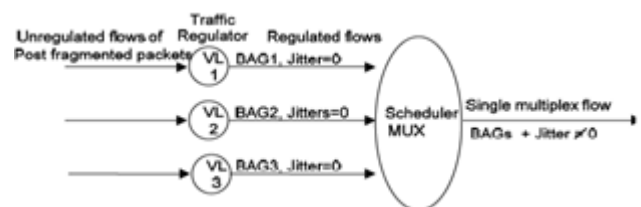


Figure7: Model of the Scheduled Flow Control Mechanism

At the output of the scheduler, for a given Virtual Link, frames can appear in abounded time interval. This interval is defined as the maximum admissible jitter. This jitter is introduced by the scheduler.

3.5 Jitter

Jitter is variation of packet arrival time after bag. In transmission, the maximum allowed jitter on each VL at the output of the end system should comply with both of the following formula.

$$\begin{cases} \max_jitter \leq 40\mu s + \frac{\sum_{i \in \{\text{set of VLs}\}} (20 \text{ bytes} + L^{\max}_{\text{bytes}}) \times 8 \text{ Bits/bytes}}{NbW \text{ bits/s}} \\ \max_jitter \leq 500\mu s \end{cases} \quad (3)$$

3.5.1 MAC Destination Address

A Virtual Link should only be identified by the MAC destination address as illustrated in Figure 8, and the MAC source address of AFDX frames should be the MAC unicast address used to identify the physical Ethernet interface. A MAC destination address in the AFDX frame should be a Group and Locally Administered address and should be compliant with the following format.

48 bits	
Constant field 32 bits xxxx xx11 xxxx xxxx xxxx xxxx xxxx xxxx	Virtual Link Identifier 16 bits

Figure 8: MAC Multicast Addressing Format

Each ES should get "constant field" and "Virtual Link Identifier" values from the system integrator. The values are not specified in ARINC Specification 664. The constant field should be the same for each ES in any given AFDX network. The least significant bit of the first byte indicates the group address (always = 1). In order to use the standard Ethernet frame, MAC group addresses should be used to send frames from End System to End System(s). The second to least significant bit of the first byte indicates the locally administered address (always = 1).

3.5.2 MAC Source Address

The MAC Source address should be an Individual and Locally Administered address compliant with IEEE 802.3. The structure of the address is specified in the following format as shown in figure 9

Ethernet MAC Controller Identification (48-bits)			
Constant field: 24-bits "0000 0010 0000 0000 0000"	User_Defined_ID 16-bits "nnnn nnnn nnnn nnnn"	Interface_ID 3-bits "mmm"	Constant field: 5-bits "0 0000"

Figure9: MAC Source Addressing Format

3.6 Redundancy Management

The Redundancy Management (RM) assumes that the network is working properly and, in particular, the deterministic properties are verified. As shown in figure10

Definitions:

- Redundant VL means that the same frames are sent through both network, A and B.
- Non-redundant VL means that (possibly different) frames are sent through either network A or B

On a per VL basis, the ES should be able to receive:

- A redundant VL and deliver to the partition one of the redundant data (RM active).
- A redundant VL and deliver to the partition both redundant data (RM not active).
- A non redundant VL on either interface and submit data from it to the partition (in this case, RM can be active or not)

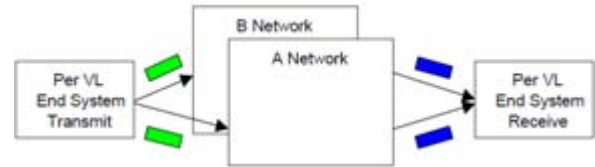


Figure 10: Network Redundancy Concept

4. Previous Techniques

MIL-1553 as its communication protocol for its projects. But MIL-1553 communication is half duplex, asynchronous and works at 1 Mbps which may not be able to cater to the future communication performance requirements of bandwidth and maintainability and delay also more. So a need exists to improve bandwidth, reliability and maintainability while at the same time reducing the physical dimensions of size, weight and number of connectors. A new network infrastructure is needed to provide more flexibility in avionics system design and cope with the increasing number of interconnections between systems.

Table 1: Comparison

S.No	Parameter	Previous Technique	Proposed Technique Using AFDX664P7
1	Redundancy	Not achieved	Redundancy achieved without errors.
2	Complexity	More	Reduced because here using Virtual Links
3	Data speed	1Mbps	100Mbps achieved.
4	Network	Deterministic	Deterministic and profiled.
5	Frame length	It not suitable for maximum frame length	Here we consider [64, 1518] means Lmin=64bytes and Lmax=1518 bytes.
6	Band width	More	Link B.W= Lmax/BAG. Here B.W reduced.
7	Type of data	Half-duplex	Full-duplex

5. Proposed Project

The proposed project is implementation of 8 -port AFDX switch on ZYNQ FPGA and also to form a deterministic network means that all the AFDX End systems connected through AFDX switches via Virtual links, finally the utilizes the data speed is 100Mbps full-duplex switched Ethernet according to ARINC-664P7 protocol. Main advantages of this project is redundancy can be achieved i.e. there is no repetition of frames at out put of end-system and another one is complexity of the network can be reduced. The tools are used - for synthesis Xilinx Vivado 2014.4 and xilinx 14.6 ISE simulator and the packet flow at end-system can be captured by Wire-shark software. The proposed architecture of 8X8 AFDX switch as shown below figure 11

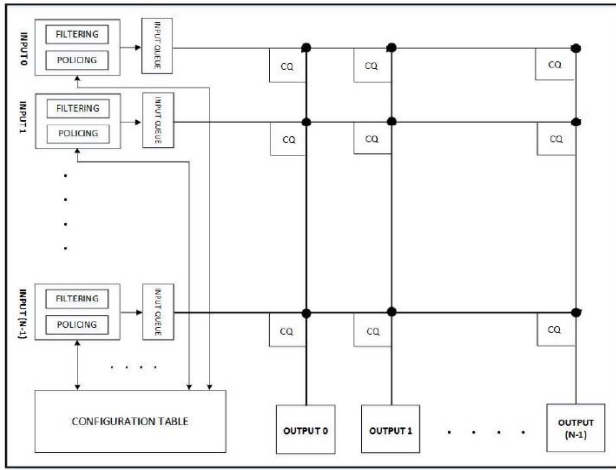


Figure 11: proposed architecture of 8X8 AFDX switch

6. Results

This section deals with the proposed hardware architecture to achieve high throughput and low latency of multicast AFDX frames through an 8X8 crossbar fabric. Input frames are first filtered and policed and checked for validity before being transmitted across the fabric. The proposed architecture will be implemented on ZYNQ board and verified for functionality. It should be noted that the architecture has been designed keeping in mind the resource constraints of the FPGA as well as the requirements of the protocol as outlined in ARINC 664-P7.

6.1 Major Requirements

The major requirements need for this project is queuing scheme, configuration table, input memory, cross point buffers and AFDX switch test setup means switches are interconnect with end systems through virtual links. A Configuration Table containing 16 entries was created with the following parameters.

Table 2: Configuration Table

Address	VLIID	Input Port	Output Ports	BAG	Priority	Jitter	Lmax	Lmin
0	1	0	3,4,5,6	16	0	10	1420	64
1	5	1	1,2,3	1	1	100	1380	64
2	7	2	5,6	64	0	40	1200	64
3	10	3	6,7	8	1	70	1400	64
4	13	4	0,1,7	32	1	200	720	64
5	15	5	1,3,5	128	0	50	716	64
6	16	6	2,4	64	0	80	324	64
7	18	7	0,1,3,4	4	1	40	623	64
8	22	0	2,3,6	8	1	90	389	64
9	25	1	0,4,7	4	0	500	1200	64
10	28	2	0,5,6,7	1	1	10	720	64
11	29	3	0,1,2,7	8	1	150	78	64
12	31	4	2,4,6	16	0	50	100	64
13	35	5	3,5,7	4	0	100	1300	64
14	40	6	0,2,3,4,6	32	1	20	1232	64
15	42	7	1,2,3	128	0	400	1000	64

To test the functionality, inputs were given as follows table 3

Table 3: Functionality Testing

Input Port	VLIID	Timestamp	Frame Length
0	1	2500	80 bytes
1	5	2000	116 bytes
2	7	500	72 bytes
3	10	1000	96 bytes
5	15	1500	88 bytes
7	18	2000	104 bytes
0	1	1602000	100 bytes

6.2 Simulation Results

The RTL was designed using Verilog HDL in Vivado Platform. Simulation was done using Vivado Simulator and the Simulation waveforms [8] are shown below.

Figure 12 shows the Configuration requests (*rqst_in*) from all inputs. It is shown in hexadecimal notation. Therefore, 04 means that the vector is $8'b00000100$, indicating that input 2 has sent a request. There is a signal named *match_found* which indicates the end of binary search, and a signal *final_addr_o* which indicates the address at which the match was found, after which we see the various Configuration parameters. The vector *conf_data_vld_o* indicates the input port for which these values are intended. Figures 13 and 14 show a detailed picture of the same.

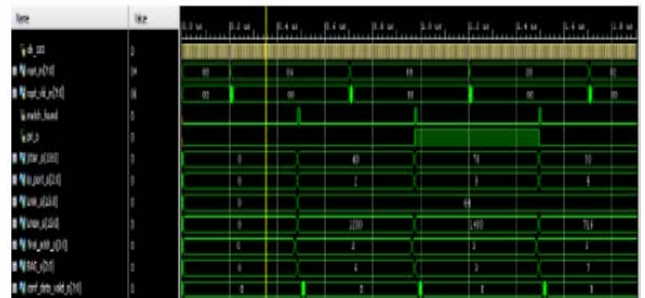


Figure 12: Configuration Table access requests and grants

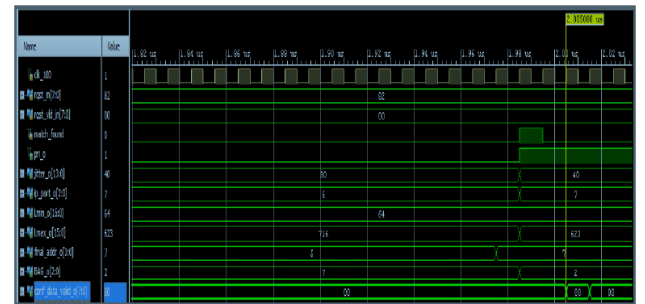


Figure 13: Configuration Table Interface

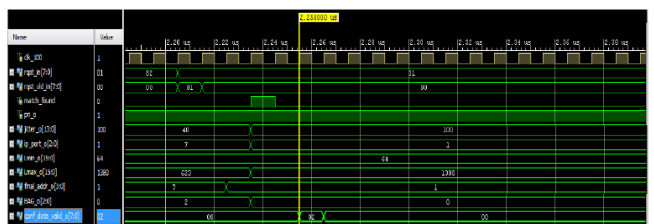


Figure 14: Configuration Table Interface

In figures 13 and 14 we can see that the request input is $8'h82$ which means that both inputs 7 and 1 are requesting at the same time. Here, the arbiter first performs the search for input 7, indicated by the value of *conf_data_vld_o* which is $8'h80$. Figures 15 and 16 shows Filtering signals for two of the input ports (input 0 and input 2 respectively). They are enabled as soon as the configuration data valid corresponds to that input. The signals *cf_vld*, *frlen_vld*, *ip_port_vld*, *mult_8* refer to constant field validity, frame length validity, input port validity and integral number of octets respectively. Once all are asserted, *filt_complete* and *filt_valid* outputs are asserted.

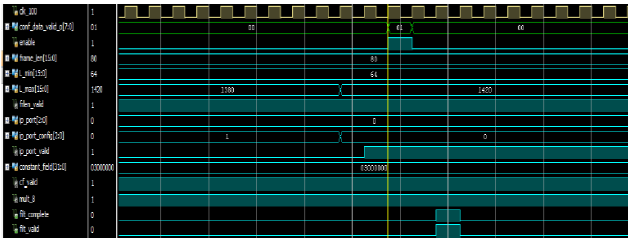


Figure15: Filtering at Input Port 0

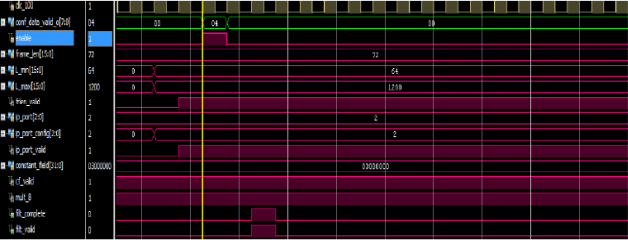


Figure 16: Filtering at Input Port 2

Figures 17 and 18 show that policing at two random input ports (input 3 and input 0 respectively). We can see that the *pol_en* signal is asserted after the configuration data is valid for that input. Following this, *ACI_vld* signal is asserted which indicates that the data from ACI Interface block is valid. The signal *first_frm* indicates that this is the first frame belonging to that VLID, in which case the frame is always valid. The ACI value to be written into ACI is (*ACI_max-Simax*). The output *ACI_new_numer_o* gives this value multiplied by 1000, and it is divided in the ACI interface block before being written into ACI memory. The validity of the numerator value is given by *ACI_new wrcmd_o*

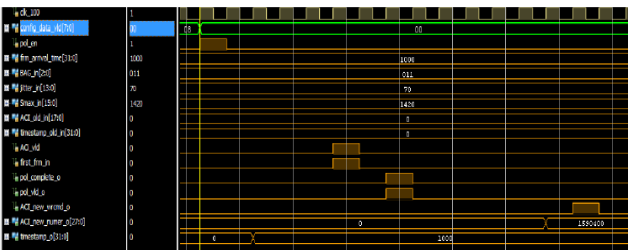


Figure17: Policing at Input port 3

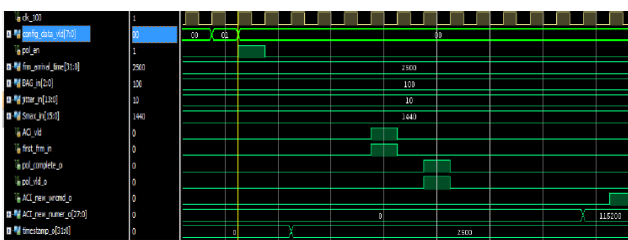


Figure18: Policing at Input port 0

6.3 End System Test Setup Results

AFDX switch was inter connect with four end systems and interface with Zynq FPGA through virtual links then to form deterministic network. For end system test set-up [9] consider VL0,VL10,VL500,VL2000 then finally we observed packet flow can be captured by Wire shark packet capture tool software and data-speed maintain at 100Mbps. Figures 19, and 20, 21 ,22 as snap-shorts of packet flow at can be captured by wire shark tool.

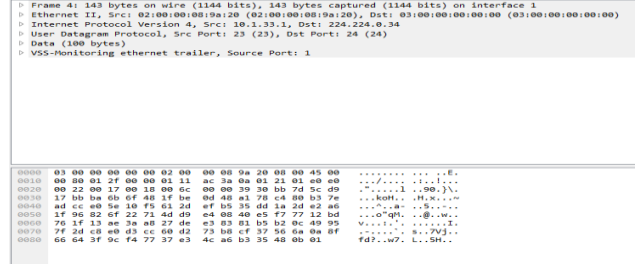


Figure19: Selecting VLLID 0 to transmit 100 bytes of data.

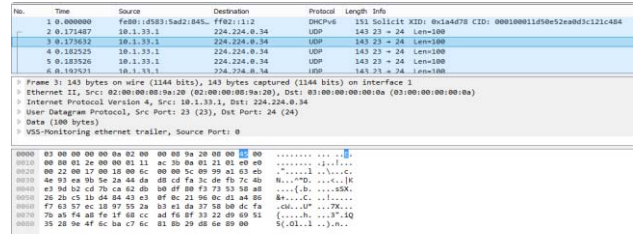


Figure20: Selecting VLLID 10 to transmit 100 bytes of data

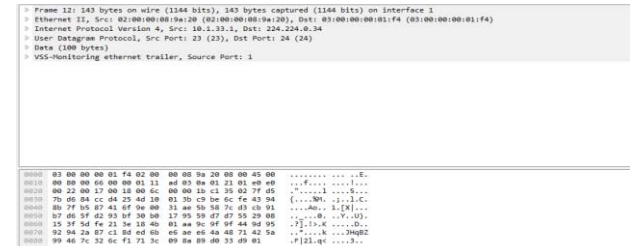


Figure 21: Selecting VLLID 500 to transmit 100 bytes of data.

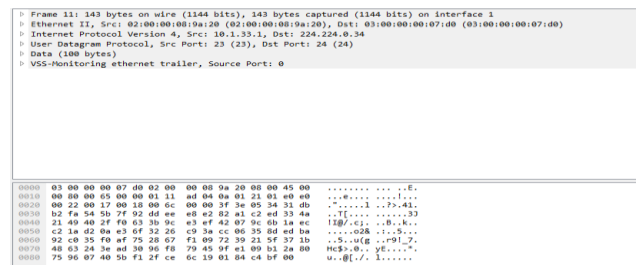


Figure 22: Selecting VLLID 2000 to transmit 100 bytes of data.

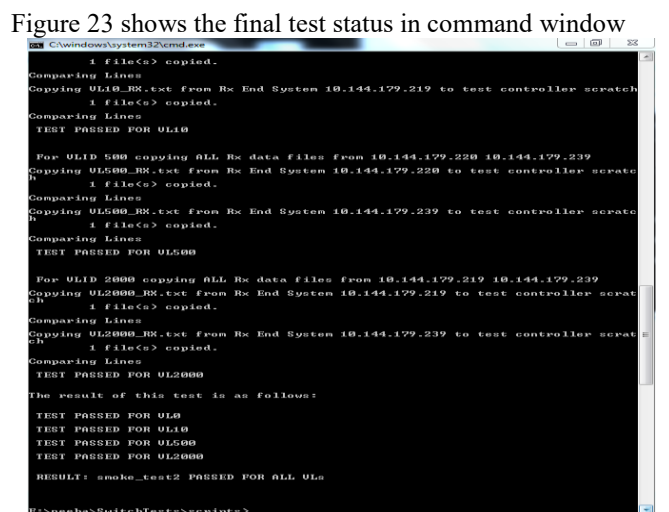


Figure 23: Test status is passed showed on command window

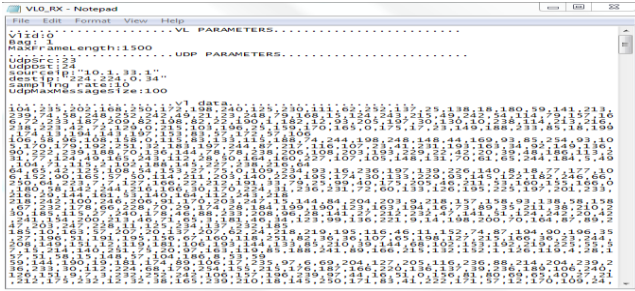


Figure 24:100 bytes of packets transferred via VL0

7. Conclusion

ARINC 664/AFDX (Avionics Full Duplex Switched Ethernet) protocol is being used as the back bone for all systems including flight controls, cockpit avionics, air-conditioning, power utilities, fuel systems, landing gear and other. Simply it mainly used in COTS, and high speed commercial Ethernet with provisions for guaranteed deterministic timing and redundancy required for avionics applications. The RTL for the proposed 8 port AFDX switch architecture was designed and simulated with certain predefined test cases for 8-port Switch. A Configuration Table having 16 entries was given based on which the traffic was filtered and policed. The behavioral simulation was found to work as desired, thereby verifying the functionality of the design and also functionality was tested by ZYNQ FPGA, by using AFDX test setup means End systems are communicate with AFDX switch via virtual links then observing data speed with 100Mbps [10] and redundancy also achieved, packet flow can be observed by Wire shark software and whole complexity of network reduced finally formed the deterministic network.

7.1 Future scope

This project is extendable to 16 ports and 24 ports AFDX switch and also it can implement on other FPGA boards for best example net FPGA and this AFDX protocol helpful in high speed Ethernet network applications it will extendable to data speed is 1Gbits/sec.

8. Acknowledgment

I am glad to express my deep sense of gratitude to **Dr.Y.Syamala**, Associate Professor, Electronics and Communications Engineering Department, for her guidance and co-operation in completing this project. Through this I want to convey my sincere thanks to her for inspiring assistance during my project. I would like thank Scientist-E **C. Ramesh reddy** at Research Center Imarat A.P.J Abdul kalam missile complex(DRDO), Hyderabad , for their support and helpful comments. I express deep sense of gratitude to **Dr. M.Kama Raju** ,HOD,ECE, Gudlavalluru Engineering college and finally I am thank ful to **Dr.A.V.N.Tilak**, Dean and Professor and P.G.Studies, for his motivation and external guidance is help-ful to complete my project.

References

- [1] Chao Zhang and Jialuo Xiao," wireless asynchronous transfer mode based fly-by-wireless avionics network", pp.1-9,IEEE-2013.
- [2] Aircraft data network PART 7,ARNIC SPECIFICATION 664 P7-1.(Base Paper)
- [3] Yu Hua; Xue Liu, "Scheduling Heterogeneous Flows with Delay-Aware Deduplication for Avionics Applications," Parallel and Distributed Systems, IEEE Transactions on , vol.23, no.9, pp.1790,1802, Sept. 2012.
- [4] Meng Li; Lauer, M.; Guchuan Zhu; Savaria, Y., "Determinism Enhancement of AFDX Networks via Frame Insertion and Sub-Virtual Link Aggregation," Industrial Informatics, IEEE Transactions on , vol.10, no.3, pp.1684,1695, Aug. 2014.
- [5] Tasnim Hamza, Jean-Luc Scharbarg, Christian Fraboul," Priority assignment on an avionics switched Ethernet Network (QoS AFDX)", pp.1-8,IEEE-2014.
- [6] Meng Li, Michaël Lauer, Guchuan Zhu, "Determinism Enhancement of AFDX Networks via Frame Insertion and Sub-Virtual Link Aggregation", pp:1684-1692,IEEE-Aug-2014.
- [7] Chakkaphong Suthaputchakun, "Impact of End System Scheduling Policies on AFDX Performance in Avionic On-Board Data Network",pp:1-6, IEEE-Aug-2015.
- [8] Nour EI-Din SafWat, Abdelhalim Zekry, Mohamed Abouelatta, "Avionics Full-duplex switched Ethernet (AFDX): Modeling and Simulation",pp:1-11, IEEE-March-2015 .
- [9] Bisson, K.; Troshynski,T."Switched Ethernet testing for avionics applications," Aerospace and Electronic Systems Magazine, IEEE , vol.19, no.5, pp.31,35, May 2004.
- [10] Henri Bauer, Jean-Luc Scharbarg, and Christian Fraboul," Improving the Worst-Case Delay Analysis of an AFDX Network Using an Optimized Trajectory Approach" ,pp:1-13.,IEEE-Nov-2010 .

Author Profile



T.Suresh has obtained his B.Tech degree in Electronics and Communication Engineering from Swamandhra College of Engineering and Technology, Narasapur in 2013 Presently he is Pursuing his Masters degree in Digital Electronics and Communication Engineering in Gudlavalluru Engineering College (Autonomous) Gudlavalluru,JNTUK,Kakinada, from 2014 to 2016. He is interested in Low power VLSI, Digital Electronics, VLSI Design.



C.Ramesh reddy has obtained Bachelor of Engineering in Electronics and Communication from National Institute of Technology ,Jamshedpur in 2002 and Master of Engineering in microelectronics from Indian Institute of science Bangalore in 2008. He joined Research Centre Imarat in 2002 as scientist, where he contributed towards realizing the Real-Time Avionics hardware and software for various strategic and tactical missile programs. His masters' thesis on Network on Chip was published in IEEE international conferences and a journal. He also holds an Indian patent on reconfigurable computing. He has contributed towards design and development of System on Chip and other Custom Integrated circuits for avionics applications. He received "Award for excellence in self-reliance for

the design of System on Chip” Currently he is working on mixed signal VLSI, Avionics Full Duplex Switched Ethernet, Multi core processor based High Performance Embedded Computing for Avionics applications, Digital Signal Processing using reconfigurable architectures.



Syamala.Y received her B.E., M.E., from Bharathiyar university, Anna university in 2001, and 2005 respectively. She obtained Ph.D from JNTUH, Hyderabad in 2014. She has been a member of IEEE, FIETE, and MISTE. She has published several papers in the area of VLSI. Her research interest includes Low power VLSI, Digital design and Testing.



A. V. N. Tilak has obtained his B.E., M.Tech. and Ph.D. from MIT Manipal, IIT Kanpur, and IIT Madras respectively. His areas of interest are Microelectronics, Digital Design, and Low Power VLSI Design. Dr. Tilak is a member of IEEE, Fellow IETE, Fellow IE(I), and Life member of ISTE.